13th ICCRTS: C2 for Complex Endeavors

"Collaborative Data Objects Enhanced Chat in Support of Net-Centric Collaboration"

Collaborative Technologies for Network-Centric Operations

Dan Winkowski, Michael Krutsch

Dan Winkowski

The MITRE Corporation

903 Enterprise Parkway, Suite 200, Hampton VA 23666

757-825-8513

winkowski@mitre.org, michael@mitre.org

This material was prepared under the FY07 Air Force MOIE Program and is Approved for Public Release; Distribution Unlimited -- Case Number: 08-0555.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2008		**00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Collaborative Data Objects Enhanced Chat in Support of Net-Centric Collaboration	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **The MITRE Corporation,903 Enterprise Parkway, Suite 200,Hampton,VA,23666**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
13th International Command and Control Research and Technology Symposia (ICCRTS 2008), 17-19 Jun 2008, Seattle, WA

14. ABSTRACT
Today, chat is being used to form collaborative communities, within and across disparate groups, (e.g., DoD, Allies, Coalition Partners, State and Local Governments, Non-Governmental Organizations, Private Voluntary Organizations) working on shared purposes towards common goals. Collaborative Data Objects (CDOs) is a new, open, standards track, technology that brings chat into a net-centric world and improves the value of this light weight ubiquitous tool. CDOs are a method of adding tailored structured data (e.g., location, target, manifest, incident assessment task) to unstructured, conversational chat without disrupting the value that chat currently provides the DoD. In addition to organizing and providing a container for data focused conversations, CDOs provide a mechanism to expose the data, and the conversations, taking place in chat following the net-centric data mandate and turning chat into a DoD information service. Finally, the CDOs technology provides a mechanism for users within chat to invoke enterprise information services through the context of the CDO data structures and defined CDO specific actions. This capability reduces cognitive disruptions from application switching, eliminates re-keying errors, allows web service results to be integrated into the collaborative discourse, and provides a means to drive external applications (e.g. GIS displays). We believe that CDO-enabled chat is the new ?command-line? of the enterprise.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**49**	

Collaborative Data Objects Enhanced Chat in Support of Net-Centric Collaboration

Abstract

Today, chat is being used to form collaborative communities, within and across disparate groups, (e.g., DoD, Allies, Coalition Partners, State and Local Governments, Non-Governmental Organizations, Private Voluntary Organizations) working on shared purposes towards common goals. Collaborative Data Objects (CDOs) is a new, open, standards track, technology that brings chat into a net-centric world and improves the value of this light weight ubiquitous tool.
CDOs are a method of adding tailored structured data (e.g., location, target, manifest, incident assessment task) to unstructured, conversational chat without disrupting the value that chat currently provides the DoD. In addition to organizing and providing a container for data focused conversations, CDOs provide a mechanism to expose the data, and the conversations, taking place in chat following the net-centric data mandate and turning chat into a DoD information service.
Finally, the CDOs technology provides a mechanism for users within chat to invoke enterprise information services through the context of the CDO data structures and defined CDO specific actions. This capability reduces cognitive disruptions from application switching, eliminates re-keying errors, allows web service results to be integrated into the collaborative discourse, and provides a means to drive external applications (e.g. GIS displays).
We believe that CDO-enabled chat is the new "command-line" of the enterprise.

Keywords: net-centric collaboration, Collaborative Data Objects, CDO, XMPP, XEP-0204, enhanced chat

Introduction & Problem

There have been several papers and articles written on the use of textual communications (i.e., text chat) as a command and control tool. Although not originally fielded as such its ubiquity in command and control and intelligence centers speaks to its value. Originally fielded as a coordination tool for system administrators (e.g. an 'order wire' for the 21st century) its ease of setup and its ease of use have made it the primary coordination and information dissemination tool in many command centers.

There has been recent research on the problems associated with text only interactions. While chat has proven value it also suffers from a lack of integration with other applications used in the enterprise. This isolation from the enterprise represents the single largest impediment for chat to support the emerging net-centric enterprise; chat as it is employed does not expose its data to the rest of the enterprise. The data in chat, the conversations representing information requests and decisions made and the topics of conversation, the actual data, are hidden from anyone who is not a member of the conversation. This exclusionary modality has forced users to start monitoring more and more rooms in order to maintain awareness in other functional areas that either supports them, or they support. A notional example of this is shown in Figure 1 below.

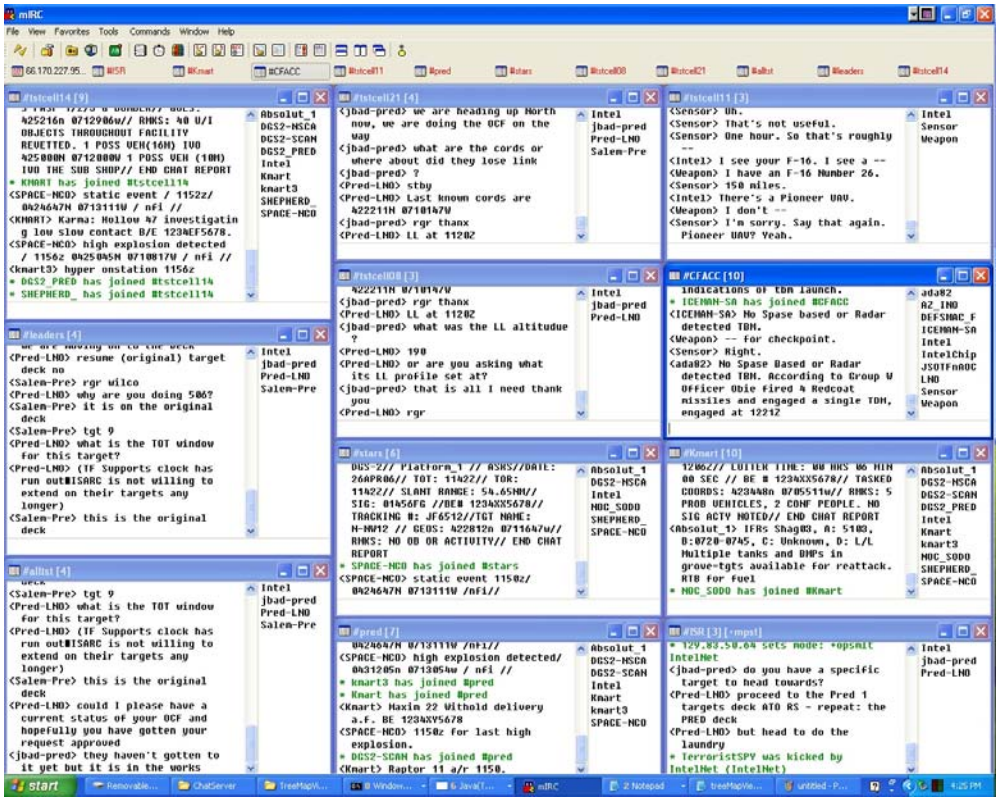


Figure 1. Monitoring Notional Multiple Chat Windows

This isolation also has a 'tunnel-vision' side effect in that as users pay more attention to chat, they pay less attention to other sources of information. The lack of chat integrated with the

enterprise means that users have to 'leave' chat in order to access or manipulate enterprise data thereby exacerbating the loss of situational awareness problem.

Another problem manifested by the lack of integration is the difficulty in getting data into and out of chat. In order to collaborate in chat using data from the many available mission applications, users often resort to selective copy and paste actions (thereby loosing any context the data had before it was transformed to ASCII characters) or they retype information (potentially introducing errors) available on one display into the chat session. Even if there were easier transfer mechanisms, Chat's lack of a strong foundation for natively supporting structured data exchanges aggravates this problem. The copy/paste integration pattern in addition to the terse, abbreviated, and idiomatic conversational style used in chat introduces ambiguity which may lead to misunderstandings and as a consequence users make faulty decisions or at least these factors impede the decision process.

Vision

Our vision for a next generation of chat is a version of chat that maintains all of its strengths (e.g, lightweight, easy to use, conversational in nature) but incorporates support for structured data without compromising its current value. Additionally, this newer generation chat would be better integrated into the enterprise so that data could be automatically injected into chat sessions to augment the interactions that occur there and results of collaborative interactions (the structured data generated or modified in these sessions) can be easily exported to systems of record without requiring users to perform complex manipulations or transformations. This integration would also provide users with access to enterprise services from within chat.

This next generation of chat would also make it possible to expose the data in chat to external users; both the topical data as well as the conversation about the topical data thus supporting the net centric mandate. Moreover, the model of having hundreds of users monitoring multiple concurrent chat sessions in order to maintain situational awareness is replaced with a model that allows users to subscribe to artifacts: created, modified, deleted, and retired and allow the subscription mechanism to alert them to changes in the environment. Users are thereby freed of the task of monitoring the conversation to find changes in state (e.g., "Are they talking about the target list? Do I have to pay attention now?").

Chat is also an isolating environment; users in chat do not have the richness of information or resources available to them that exists in the enterprise applications. Chat, enhanced by structured data, would allow the users to invoke methods on the structured data that would in turn invoke enterprise services providing the user in chat with a reasonable subset of information that would support the current task.

Under a MITRE Technology Program (MTP) sponsored Air Force Mission Oriented Investigation and Experimentation (MOIE) research project we investigated the following hypothesis; we could overcome some of the disadvantages of text only communications and add additional value to chat if we could find a way to marry structured data with the unstructured conversation stream. While there is a different ongoing body of research in teasing useful data

out of natural language, we believe that a nearer term solution would provide for the representation and exchange of data in a structured manner. This structured data, if interleaved as part of the unstructured chat conversation, would provide context and reinforcement to the collaborative dialog. Representing data in a structured manner might provide the current benefits of being able to maintain situational awareness, or get necessary 'tip-offs' without resorting to reading a conversation line-by-line thereby reducing some of the current cognitive overload.

The addition of structured data as part of the conversation should support a full range collaborative operations rather than be limited only to viewing the data. Further, the addition of structured data should introduce novel mechanisms to enhance chat. The challenge to this vision would be to provide the aforementioned capabilities without destroying chat's convenience and ease of use.

Investigation

While technology transition, and commercial adoption are not the usual starting points of most research and development, we considered them important goals if our research was to have the intended impact. Consequently, many of the early architectural and design decisions were made with these goals in mind. We started with a review of chat standards and choose the eXtensible Messaging and Presence Protocol (XMPP) described in Internet Engineering Task Force (IETF) Request For Comments (RFC) 3920, 3921, 3922, and 3923 because of its maturity of standards, ease of extensibility, and support for structured data. XMPP is a family of standards with one standard describing a transport layer and another describing a presence and messaging overlay. The XMPP Standards Foundation (the protocol's governing body) also has a process for validating, accepting, and managing extensions to the base standards so that additional functionality can be layered on the core standards in a manner that is still open and non-proprietary;the core standards are extended through a community standards process. An independent software foundation and an executive council oversee proposed extensions to the core standards. Peer review of the proposals ensures that extensions have broad appeal and are free for all parties to use in their development. An added benefit to our work is that XMPP is the mandated chat protocol for the DoD. With the availability of usable specifications, open source XMPP implementations, and a mechanism to contribute our work back to an open community, we were able to concentrate our efforts on the design, presentation, and operations over structured data.

Almost all multi-user chat follows the client-server design pattern. Users send chat to the server and the server takes responsibility for distributing the chat to all of the members of the chat room. If a user is disconnected and reconnects, the server has the responsibility of sending the reconnected user all of the conversation that occurred during the period of time that the user was disconnected. This model suited our needs for structured data. We would make the server responsible for synchronization of the data across all of the members of the conversation. This design decision would require extending the notional behavior of the server to accommodate distribution and synchronization of the structured data as well as distribution of the conversation(s). Recognizing that, in the DoD, many chat users are on disadvantaged networks we decided that we needed to find a way to send the smallest amount of information possible in

order to conserve bandwidth and this would probably require the clients to be smart about the information they sent to the server. These design choices are reflected in the architecture figure below:

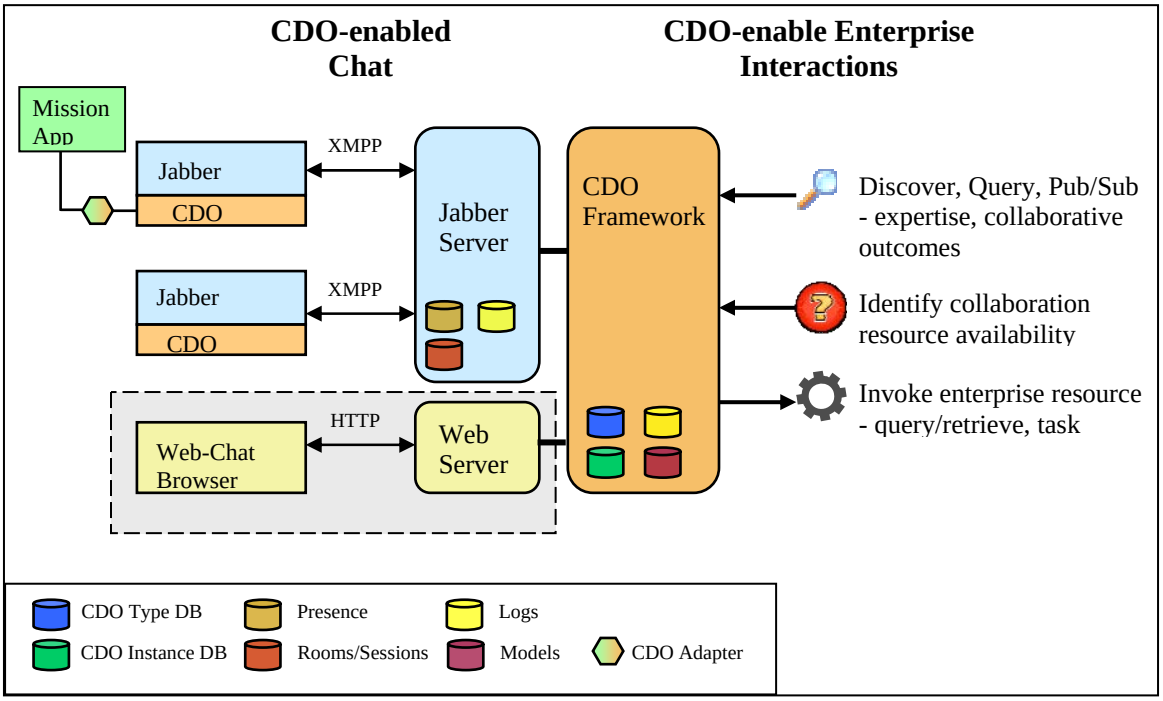


Figure 2. CDO Architecture

We assumed that data in the CDOs would change over the course of interaction (e.g., User Bob set status from 'planned' to 'final') and we wanted to be able to record the changes and attribute the changes to the author. Moreover, we wanted to be able to propagate only the changed data thereby preserving as much bandwidth as possible. We investigated existing XMPP support for structured data exchange (i.e., XMPP Extension Protocol (XEP)-004 Data Forms) but did not find the level of versioning and support for collaborative changes needed. Collaboration over CDOs would involve not only creation but update and deletion as well. Therefore a protocol supporting robust synchronization and versioning would be required.

Another design objective was to separate the description of the data from the framework that would support it. Therefore, a client receiving information about a particular subject would need only to load the appropriate CDO type to operate in that data domain context. The same architecture that supports the DoD in mission planning and execution could also support a homeland scenario such as disaster coordination as well as common commercial practices (e.g., financial institutions with structured buy and sell orders). Currently, Wall Street traders conduct trades using simple chat without significant support for structured data. Lack of structured data support requires the traders to read and parse all messages, interpret the contents and make decisions about what transactions to pursue. A typical interaction might have a seller typing in information about something they might want to sell and waiting for a barrage of questions like: how much do you have to sell, how much do you want, when do you want to sell, will you sell part or is it a package deal, etc. In CDO enabled chat, the seller would instead inject a CDO into

the trading room populated with the relevant information (e.g., name, stock ticker, current trading price, number of shares to sell, asking price, single sale or partial sales, etc.). In this manner buyers would have all of the information they need to make their "buy or pass" decision contained in the 'sell offer' CDO. Moreover, under this scheme, a buyer could have a software agent easily reviewing sell offers since the contents of the sell offer are well described by the CDO's type description and the information provided in the CDO would allow automated review and forwarding (to the human) of offers that met the trader's particular criteria.

In summary, the initial design requirements established called for 1) building off of the XMPP set of standards, 2) a domain agnostic description language for defining CDO types, 3) a new XMPP protocol for exchanging encapsulated data structures and changes to the structures that result from user collaborative updates, 4) a CDO client and server framework for managing CDO content and user interactions over CDOs, and 5) appropriate abstraction in design so that neither client nor server code would require changes to adapt to new data contexts. However, in order to support the full scope of the vision outlined two additional design objectives were introduced. 6) CDO content should be accessible (exposed) to the enterprise following the tenants of the DoD Net-Centric Data Strategy. A consequence of this objective is that the CDO architecture would need to interoperate with DISA's Net-Centric Enterprise Services for Content Discovery and Delivery. 7) Consistent with DOD's move towards a service oriented architecture (SOA) and program's establishment of new information services, Collaborative Data Objects require the addition of action semantics to enable user interaction, via a CDO context, with external information services.

NC Capabilities

In this section we introduce the CDO capabilities developed during FY06 and FY07 as part of the MITRE Research Program.

CDO Description Language

The project developed a CDO Description Language (CDO-DL) for declaratively specifying the design a CDO type. A CDO-DL consists at minimum of: a structured data design, a presentation description section, and may also contain a set of 'methods' (user accessible actions) like: execute a predefined query, plot an object using a mapping application, or to execute a search based on the contents of the CDO.

For the data design portion we chose the W3C XML Schema language. The choice allows for explicit description of a CDO's structure, CDO instance validation, and lays a foundation to support future exchange of structured chat data across security domains. For the presentation description we selected the W3C XForms specification. XForms is used to declaratively describe the presentation layout so that the structured data can be both visualized and manipulated in a human friendly (and device independent) manner thus facilitating collaboration over the structured data. The combination of these technologies allowed us to define a CDO description language (CDO-DL) that would provide us with plug and play CDO types. Plug and play CDO

types affords us a domain agnostic solution; the architecture doesn't care what kind of data is in the CDO and all CDO's are fundamentally self-contained and equal.

Treatment of the CDO-DL method description capability will be introduced later in the paper. The CDO-DL, sans method description, is documented in the CDO XMPP Extension Protocol (XEP) XEP-0204 as submitted to the XMPP Standards Foundation (XSF).

Data Synchronization Protocol

In developing the protocol, we were mindful of the need for a lightweight style of synchronization capable of working in limited-bandwidth conditions. The (near) tactical edge user operates in environments where bandwidth may be restricted and latency excessive. We investigated solutions that would extend the base XMPP transport and messaging protocols. We reviewed current extensions to the core protocol for Data Forms (XEP-0004) and Publish and Subscribe (XEP-0060) and found that our requirements for data validation, manipulation, and data distribution exceeded the scope of these extensions. DoD edge users typically do not have the bandwidth budget to afford high volume data transfers with redundant information. We needed a synchronization model that would allow the end user to determine the update rate and keep the updates as small as possible while placing the synchronization burden on the software and the system.

A number of additional synchronization schemes were investigated. Among these were the (then) proposed Simple Sharing Extensions for RSS and OPML. While not a perfect fit for our needs it did provided inspiration for our protocol development. Our data exchange and synchronization protocol was designed to would work in concert with the XMPP messaging and transport protocols to support full or partial (field level) data object validation and synchronization. Extending the core XMPP protocols through the XSF's open process will allow us to easily transfer the results to industry and prevent proprietary implementations that will inhibit its uptake.

The result is an experimental XEP (XEP-0204) named Collaborative Data Objects that describes the data synchronization protocol. A feature of this protocol includes the synchronization of changes to a CDO instance all the way down to the field level using either a strict or lazy synchronization approach. Under strict synchronization, every change is pushed to a client but under a lazy model the client is only notified that his local copy is out of date and that client is responsible for requesting (polling) for updates from the server. Operations at the CDO instance level include create, update, and retire of CDOs, while operations supported at the field level include create, update, and delete. The protocol also describes not only how CDO instances are synchronized across clients but also how administrative procedures related to the exchanged of CDO types are handled. Clients may request and retrieve from the server a CDO type instance that they do not have loaded. Most often this is in response to the receipt of a data synchronization message for a CDO type that they are unable to interpret since it had not been pre-loaded on the client.

The data synchronization protocol lays the foundation for structured data collaboration within chat; every thing else builds upon it. MITRE's intellectual property for the data synchronization protocol has been transferred to the XSF and, as has been previously mentioned, this protocol has been published as Collaborative Data Objects XEP-0204.

Client and Server CDO Framework

We built a reference implementation of the protocol as a proof of concept by developing a Collaborative Data Object framework that is used to augment both XMPP servers and clients. Building a proof of concept allowed us to validate the protocol in addition to having a demonstration system to gain user feedback on the concept. We took advantage of the plug-in interface that is available in most modern XMPP servers to give us access to the XMPP message stream. This message stream interface allows the framework to detect and manipulate those message stanzas containing CDOs (it ignores message stanzas whose content does not include a CDO) without having to modify the XMPP server. Correspondingly we took advantage of the plug-in interface of an open source XMPP client to embed additional capability in the client (e.g., create, view, update, retire, and visualize CDOs).

The existing demonstration capabilities allow users to use the XMPP client to manually create and manipulate (see Figure 3) any CDO type appropriately described by the CDO description language. Several CDO types have been created for experimental purposes.

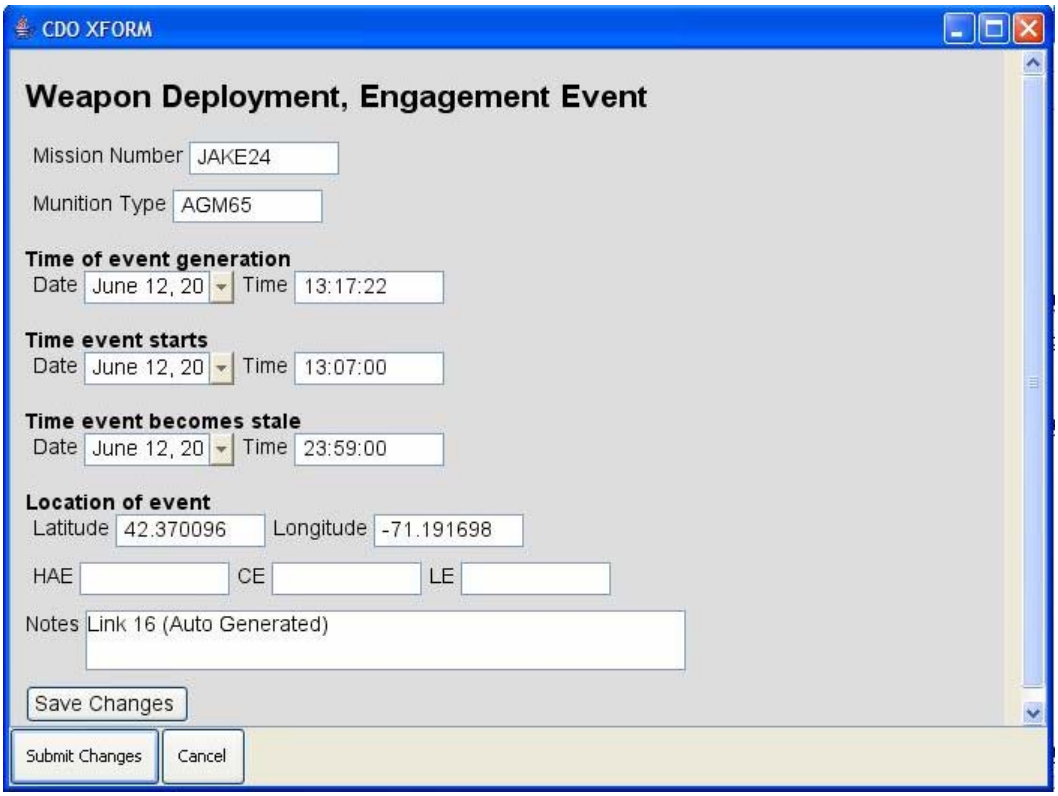


Figure 3. Sample Weapon Deployment CDO

Once a CDO instance has been created or updated via the XForm a user will submit the changes. This submission action transmits the CDO to the server-side framework for validation and distribution via an XMPP message stanza. Post validation, the server-side framework will send a copy of the CDO to either: a) the other party in a pair-wise Instant Messaging session; or b) all of the members of a multi-party chat session. At the client, receipt of a CDO message event is depicted as an activation token that closely resembles a standard hyperlink. Instead of the familiar HTTP scheme a new CDO scheme is employed to denote a URL to a local CDO cache on the client. Additionally an icon is prefixed to the URL to complete the rendering of the activation token (see Figure 4).

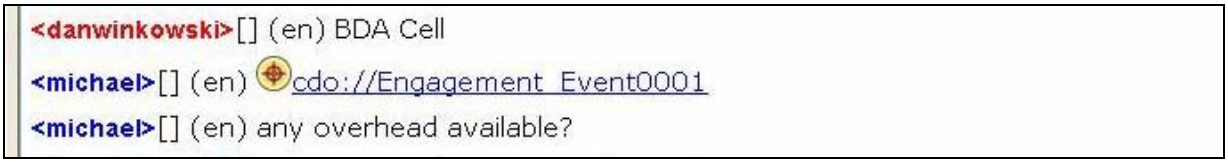


Figure 4. Sample Activation Token

While there is definite value to being able to manually create new structured data within chat, user feedback has indicated the greatest value comes from the ability to allow external applications to inject selected data automatically (i.e., save users from the chore of manual data entry either by copy/paste actions or by retyping) into chat. Similarly, the ability for a user to automatically transfer a completed CDO instance from the chat context to an external application is equally valuable. During FY '06 the client-side framework was extended with application specific adapters to illustrate the concepts of moving data into and out of chat. An import function, similar to a "Paste As …" function allows the XMPP client to manually import the contents of a serialized XML file that might have been exported from an application; future work will support seamless transfer of CDO instances between chat clients and applications.

Beyond the basic capabilities described to this point, more advanced CDO features were developed in FY' 07. Among these is the introduction of CDO Methods that will allow actions to be invoked over a CDO instance. Methods enable an external resource (e.g. web service, application API) to be called according to a prescribed design pattern using the field contents of a CDO instance as parameters. Typical examples are making a query based on the contents of a CDO (e.g, query a database for assets within a geographic location described by a latitude/longitude bounding box in the CDO) or issuing a Common Alerting Protocol (CAP) message based on information contained within the CDO. Shown below is a contextual menu associated with the CDOs' activation token allowing the user to choose actions appropriate to the CDO (See Figure 5).

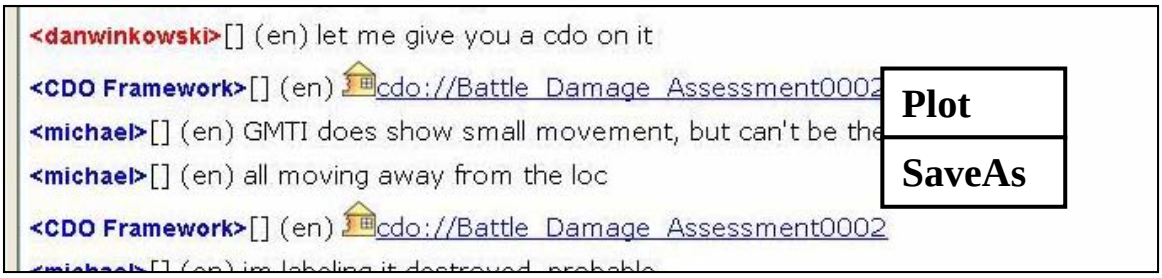


Figure 5. Sample Contextual Menu

As shown, one of the actions explored was to plot the contents of a CDO (which has location information like a latitude and longitude) onto a map (see Figure 6). The plot action was bound to a Google Maps API call and the appropriate location fields within a test CDO type.

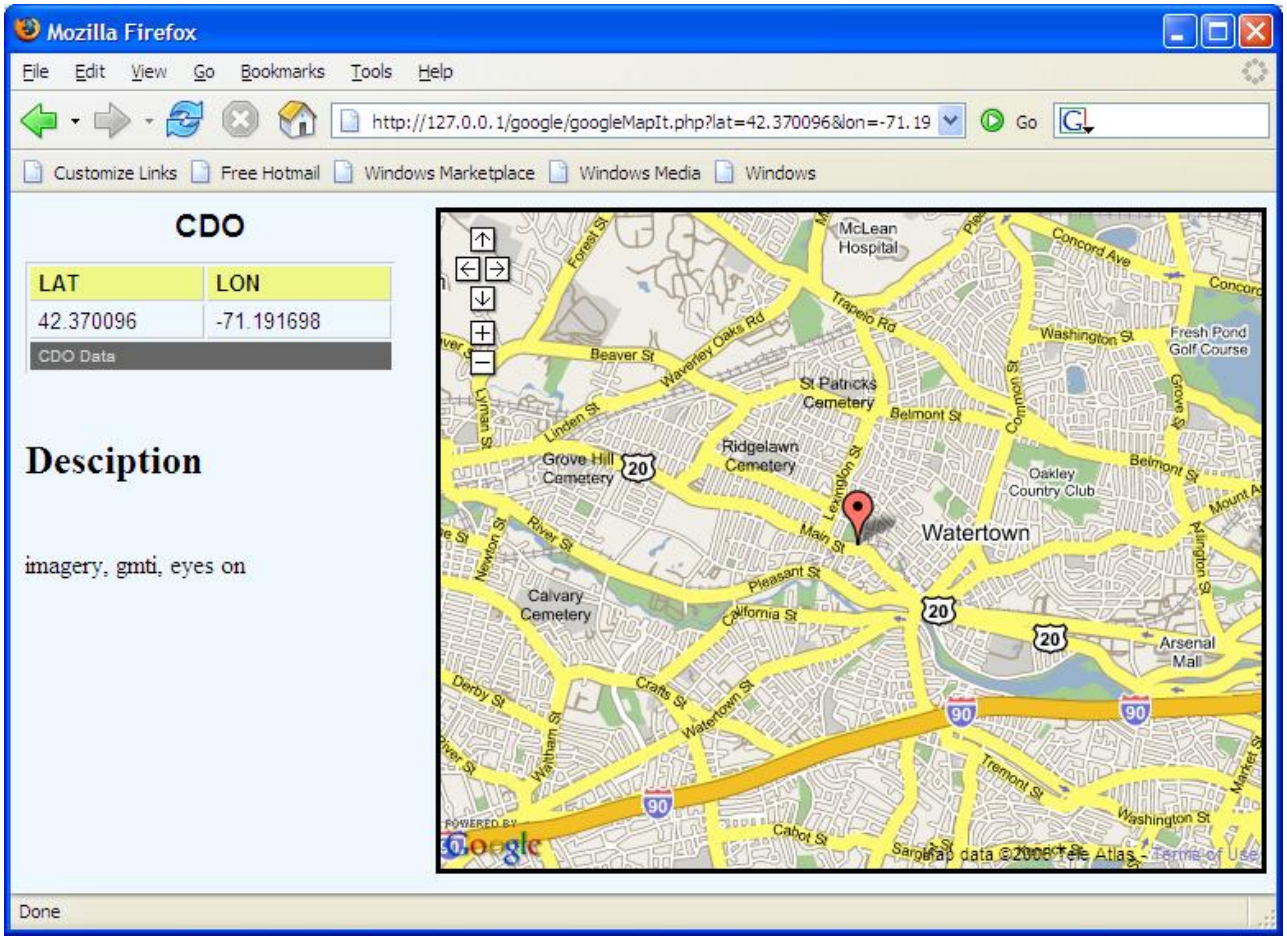


Figure 6. Sample External Application Support

Exposing collaboration spaces to the enterprise

The research successfully developed and demonstrated technology to advertise CDOs according to then DOD Discovery Metadata Specification (DDMS). It was shown how CDO types could be mapped to a general taxonomy and how each CDO instance could be translated to a DDMS advertisement. A prototype DDMS adapter was developed as proof of concept demonstrating integration with the Net-Centric Enterprise Service (NCES) Content Discovery Service (CDS). The adapter periodically queries the CDO transaction (instance) database to determine the state of newly changed CDO instances. A CDO type specific XSL style sheet is applied to generate the matching DDMS advertisement which is then published. The resulting capability provides enterprise participants the ability to query and track collaborative chat products in the same manner as any other information asset.

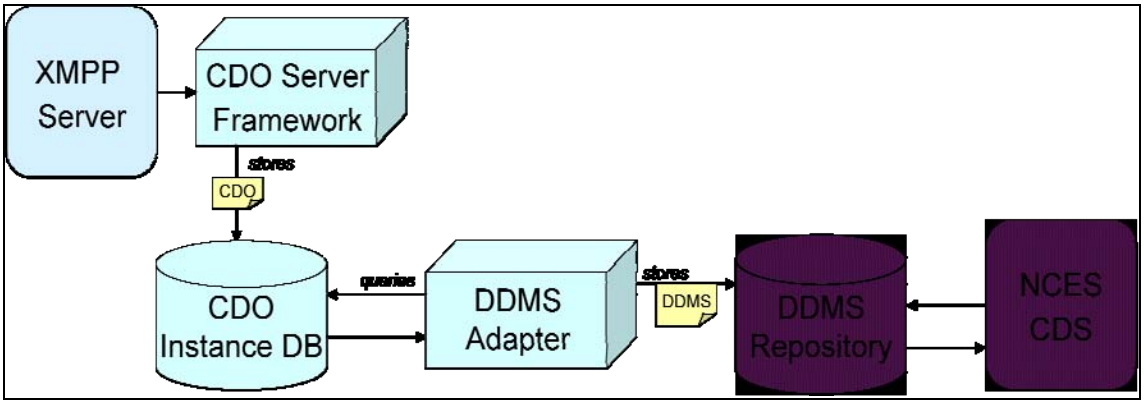


Figure 7. CDO-to-Enterprise Integration

Another of the objectives of the research was to provide a means to support finding subject matter experts within chat rooms. CDO DDMS advertisements enabled this capability. A CDS query can be constructed to find expertise based on contributions to various information products. The expert can then be contacted through his chat identifier (made available in the CDO DDMS record). A more sophisticated expertise finding capability is possible by applying data mining techniques to the CDO transaction database which records every change to a CDO field and the author of the change.

It should be noted, that due to lack of access to an experimental NCES CDS capability, a substitute capability was used during development. Specifically, an Army funded, MITRE developed, Data Dissemination Service (DDS) was installed in our lab. DDS adheres to the DDMS specification and provides a query and pub/sub capability based on DDMS advertisements. Therefore DDS was used as a proxy for CDS in all lab tests.

In addition to the DDMS adapter the MOIE constructed a Really Simple Syndication (RSS) adapter as well. RSS is an information syndication technology that is commonly used on the internet. This RSS adapter creates an RSS feed for each chat room so that users can subscribe to the feed and receive updates on changes to CDOs in a room. This was demonstrated using an unmodified RSS client.

Finally, it is important to understand the power of the content discovery paradigm coupled with CDO technology. The NCES CDS today provides a portal through which information assets can be queried and retrieved. Queries are conducted using a web form and results are displayed via an HTML page. The web query form itself can be represented instead as a CDO type (since it consists of a set of data fields used to compose a query). Users could use such a Content Discovery CDO to collaboratively define their information requirements and to retrieve information directly into chat by executing a query against the CDS information service (see next section). The MOIE has in fact succeeded in demonstrated this capability.

Provide chat access to enterprise capabilities (information services)

The MOIE developed new technology to support chat/enterprise integration. Specifically, a declarative based CDO Method Description Language was developed that forms the foundation for a general and flexible approach for defining the variability found in interfaces with enterprise information services. The language describes the correspondence between CDO data fields and parameters required for an information service call. It also addresses interaction patterns in the areas of 1) User input; 2) Method call type; 3) Service result types; 4) Output handling; and 5) Data routing and transformation.

An example may help to demonstrate the power and versatility of this language. One type of CDO developed by the MOIE was for meeting coordination. The Meeting Request CDO consists of fields for the meeting title, start and end dates and times, meeting room, and participant email addresses. Such a CDO may be useful for coordinating in real-time the particulars of a meeting. In fact such real-time coordination may have distinct advantages over lengthy multi-party asynchronous coordination typical of email. Now, suppose a company provided a room reservation web service which required the parameters start and end dates and times, number of participants, and campus for hosting the meeting. A method to describe the Meeting Request CDO's use of this service could use the following interaction patterns in its definition (see figure 8 below):

- User input: prompt for missing parameters (note the Meeting Request CDO does not supply a field for campus).
- Method call type: Discover Properties of Identifiable Resource
- Service result type: List (meeting rooms matching the criteria)
- Output handling: Single Value Select from List Dialog
- Data routing and transformation: Bind Selection to Field (Meeting Room)

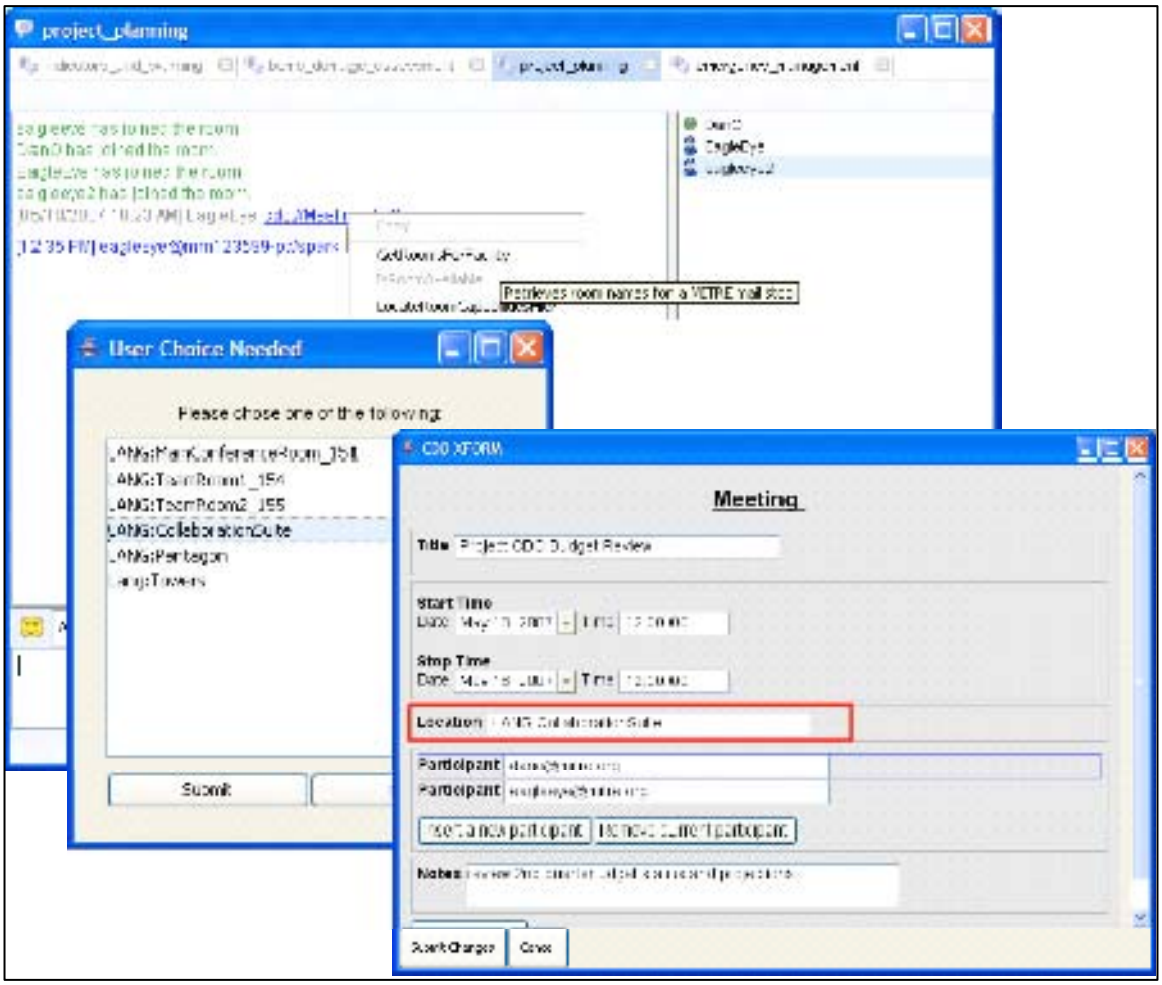


Figure 8. Example Information Services

A user invoking (menu accessed by right mouse click off a CDO) this action would be prompted for the missing parameter (the campus, not shown in figure) and then be presented a list of suitable rooms returned by the web service. Selection from a list dialog would then update the meeting room field of the CDO. Given the appropriate web services a method description modeled along similar lines could be used to identify the availability of a runway in the context of mission planning or the availability of aircraft in a dynamic re-tasking scenario.

Interpretation of a method description is handled by the CDO Method Invocation and Binding Framework technology. Under this framework, the chat client is not responsible for actual invocation of the information service; rather a separate method dispatcher receives a method request from the client, executes the call, and returns the results to the client for processing. Supporting a separate dispatch mechanism promotes flexibility in terms of integration with future middleware and reduces the complexity of the client. This generality promotes loose coupling as service endpoints may vary. Likewise the declarative approach to CDO method description provides for plug and play deployment of methods onto clients since no client modifications are required.
The steps involved in a method invocation are shown in figure 9 and briefly described below.

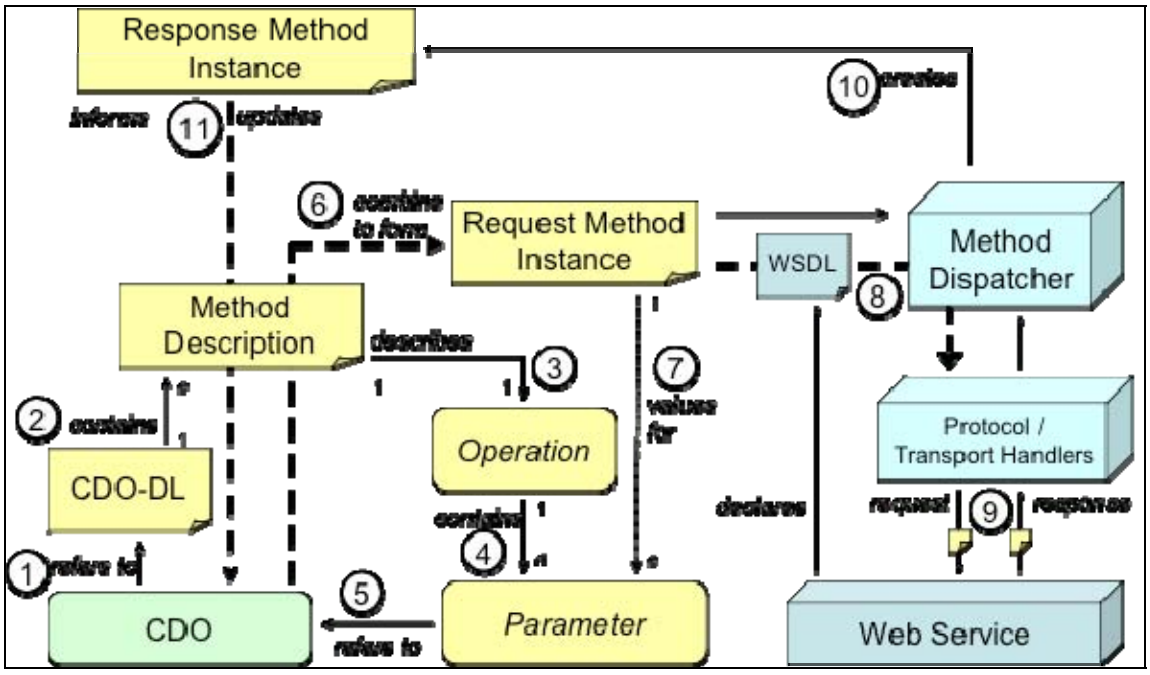


Figure 9. Information Service Detailed Architecture

1. A CDO instance refers to its CDO type declaration that is specified using the declarative CDO Description Language (CDO-DL).
2. CDO types contain a section for the type's Method Descriptions
3. Method Descriptions describe a single operation
4. Operations contain zero or more parameters
5. Parameters can point to the content of a CDO (e.g. parameters for an operation may refer to CDO data fields)
6. When a method description is invoked, the current content of the CDO is used to create a method instance describing the request. The User Input section of the Method Description governs this behavior. The client implements a library to execute the various documented User Input interaction patterns.
7. Request method instances contain values for every parameter
8. Method Dispatcher uses the content of the request method instance and the target's service WSDL to determine the protocol & transport over which to invoke the service. It should be noted that there a practical limits to the variety of CDO/information service interactions. These limits are documented and described as Method Call Type patterns.
9. The service is invoked, and a response is received.
10. Method Dispatcher creates a Response Method Instance. It assures that the service response conforms (or is translated) into one of the supported Service Result Types (string, list of strings, reference (URL), or CDO instance).
11. Response Method Instance is used to update the CDO or present information to the user. Specifically, the client implements an interaction library to handle the response according to the Output Handling and Data Routing and Transformation interaction descriptions (described in the Method Description).

CDO Interaction Patterns

As previously noted, the CDO-DL language's method interaction section follows a pattern based model for describing the variations for CDO enabled user interaction with information services. The research team has implemented (bold) a number of the patterns identified. These are listed below to provide the reader a summary overview of the supported patterns.

User Input
- **Menu visibility when parameter preconditions met**
- **Parameter prompt**
- **Method call** or cache access

Method Call Type
- **Discover properties of identifiable resource**
  - **With Range or Property restriction**
- **Property value retrieval**
- Property value set

Service Result Type
- **Content Reference : URL,** CDO
- **Discrete Primitive XSD: Simple Type**
- **List**
- Tree
- **Complex type**
  - Binary (mime-type handling)
  - **Structured data**

Output Handling
- **Browser Display of URL**
- **Screen Echo**
- Confirm/view dialog
- **List selection (single, multiple)**
- Tree navigate, select leaf
- Tree navigate, select branch
- CDO reference

Data Routing and Transformation
- **Transform**
- **Transmit - send to chat as text**
- Negotiated method refinement
- **CDO item** update/**create**/delete
- Store locally to named cache

Human Intelligence Task (software calls people as a service)

An unexpected result of this research was the identification of the Human Intelligence Task (HIT) pattern. Amazon uses this pattern in their Mechanical Turk web service suite. It provides a

way for applications to avail themselves of human expertise to perform tasks that an application can not. Within the chat context, this pattern turns out to be the reverse of operators invoking information services from within chat. In effect it provides a way to treat chat rooms (or at least the resident experts) as an information service.

The CDO technology provides the basis for implementing the HIT pattern within chat. Using this approach an application would inject a CDO representing the information being sought into the appropriate chat room. Operators would examine the request, collaborate, and record their results by completing the missing fields in the CDO. Task completion could be indicated either by a special field monitored by a software agent or through a method to call back to the calling application. The HIT pattern was successfully demonstrated by the researchers. In the scenario demonstration we envisioned an application monitoring CNN closed captioning for disaster related incidents. The application requested human expertise by injecting an Incident Assessment CDO into the emergency management chat room. Operators would review the linked video and complete the CDO fields (which followed the Common Alerting Protocol) to enter their evaluation. On receipt by the application the evaluation results would be structured for further machine processing.

Current practice in military operation centers (e.g. AOC) employ chat procedures that describe the formation and manning of functional rooms tied to operational workflows. This capability may be used to enable numerous application-to-human workflows within such organizational constructs. The potential benefit is increased speed of operations by reducing machine to human to machine impedance mismatches and realization of SOA benefits by allowing applications to employ collaborative expertise of operators through a service based interface.

Assessments

The CDO MOIE participated in several assessment events.

In November 2006, the Global Cyberspace Integration Center (GCIC) Integration Task Force (ITF) completed a detailed analysis of the CDO concept and technology. They mapped the technology to applicable warfighter requirements related to 1) providing seamless, tailorable, and easily accessed worldwide C2 communications and 2) to improved chat technology to support integrated space planning, support for day to day and crisis operations, global ops/intel monitoring for situational awareness, and operations integration and synchronization. The assessment included a set of scores which judged CDO over a number of dimensions. Score highlights included ranking the operational value as a *unique capability, enables other programs,* the investment value as *moderate pay-off, good bang-for-buck,* and the net centricity categories at the highest end of the ranking system. As a result the ITF directed the project to engage with the AF Transformation Center (AFTC) on future experimentation, assessment, and warfighter alignment.

In January 2007 we participated in a GCIC subject matter expert (SME) event where the CDO concept and technology was briefed to a diverse audience of subject matter experts. The event provided a vector check for the project and identified a number of use cases for the technology.

In June 2007, CDO was assessed by MITRE staff on behalf of USSTRATCOM. They reaffirmed the relevancy of the technology and recommended that, once it becomes an XMPP standard, the capability be included in a future version of the NCES collaboration suite.

On 26-27 September 2007, the CDO project participated in a USSCOM warfighter workshop conducted in conjunction with MITRE and the GCIC. During the engineering event the GCIC Modernization & Innovation division conducted an operational evaluation of the CDO technology on behalf of the GCIC ITF. Staff from Air Combat Command (ACC) and the GCIC manned three stations in support of an air operations planning scenario using CDOs. While the workshop provided limited opportunities to demonstrate the full range of CDO interactions, particularly with web services and the HIT pattern, the base CDO technology (collaboration over encapsulated data objects via chat) capability was evaluated and the results summarized below.

Operational Findings: Overall CDO tool operated well and provided a new capability that is a more efficient way to handle collaboration and C2 in a text chat environment. The event only provided a limited look at the overall potential of CDO's since this capability has the potential to be used in a wide variety of Warfighter processes. The CDO's required very limited training for operators and were quickly incorporated into the operator's collaboration process. Data previously limited to text only within chat was incorporated into logical structures which could be updated, edited, archived, and made available to all enterprise chat participants speeding operations and reducing confusion and error, while contributing towards DoD's move to a network centric data strategy. From an operational user interface perspective, future development is needed to enhance the ability of operators view and manage large numbers of CDOs. A summary type overall interface is needed so operators do not need dig through large amounts of text chat logs to find and track CDO's that are being worked.

Recommendations: Overall operator feedback was very positive. The CDO has a potential of enhancing collaborative capabilities and merging non-structured and structured environment with the enterprise. Operators believe this capability is of great value and should continue to be matured. This technology should be looked at in future workshops, exercises, JEFX events and other venues events to expand the Warfighter feedback and mature the capabilities and TTPs.

Conclusion

The CDO MOIE's impact can be evaluated in part according to the original research question. We examine this question in two parts. First we posed the question "*Can chat, enhanced with structured collaborative data objects, overcome the human-enterprise conversational barrier …*"

During the course of the research we have shown that CDO enabled chat can be the mediator for human enterprise interaction. Enhancing chat with structured objects provides the foundation for human-enterprise conversations to take place in a manner understandable to both operators and applications. Both CDO's use of the Human Intelligence Task pattern and the method invocation capabilities are examples of overcoming the human-enterprise conversational barrier.

The second part of the question "*... to support seamless integration between users and the enterprise via the chat paradigm?"* can be answered in the affirmative as well. This statement is supported by the following observations:

- Data previously stove-piped within chat can now be made available to all enterprise participants.
- Operationally, this means that users can search & monitor collaborative outcomes without lurking within chat.
- The number of rooms that need to be actively monitored by operators can be reduced and data previously locked up in chat is now available anytime, anywhere.
- Chat operators can directly access enterprise information services through CDO technology according to domain/mission context.
- Operationally, cognitive disruptions can be minimized and errors due to data re-keying reduced.
- Faster information access, collaboration, and self synchronization become possible.
- The Human Intelligence Task interaction pattern links mission processing needs to collaborative expertise.
- Operationally, mission workflows can be accelerated by employing familiar tools to access operator expertise returning added value back to the workflow.
- If mission services can directly request expertise resident in chat rooms, chat rooms become in effect an enterprise information service.

Another measure of the MOIE's impact is by gauging its influence and transition success. The project's transition strategy was always two fold. First, we sought to develop and shape future chat standards so that the technology would become available commercially. Second, to educate DoD sponsors and programs on the benefits.

Portions of the CDO technology have already been transferred to industry. MITRE released its intellectual property rights on the core CDO data management and synchronization protocols and submitted them to the XMPP Standards Foundation (XSF). The protocol was accepted as a standards track XMPP Extension Protocol (XEP-0204). This is the official process through which new XMPP standards become recognized following a peer review process. Currently, the XEP is in the experimental stage.

MITRE open sourced its FY06 and FY07 reference implementations. Reference implementations help the XSF community not only to judge the feasibility of an idea but also to build a base of support and mind share. Additionally, throughout this period the project has engaged not only the XSF but also the vendor community. XMPP vendors such as Jabber Inc., Jive Software, Coversant, and Google have been briefed on the technology. IBM, the developer of a non-XMPP chat technology was also briefed since the project believes that the CDO capability could be implemented over other (non-XMPP) chat protocols.

Lessons

The project has learned valuable lessons ranging from use of new and emerging technologies through valuable input during user assessments. We though it was very important to use technologies based on standards or to extend the standards and submit our work to the appropriate standards bodies. In this vain we chose XForms, an emerging W3C recommendation, for our visualization component. While we still believe this to be a good decision, the lack of mature, fully capable open source implementations caused us to make choices and expend resources that we would have preferred not to make. This is simply a function of working in the research world with emerging technologies.

We also believed that there was value in exploring advanced concepts with unexploited synergies like collaboration and Service Oriented Architectures. Collaborative technologies are often seen as more modern replacements for the time-honored telephone instead of information services to be explored as we did in the Human Intelligence Task. While tying these technologies together has been valuable from a research perspective the difficulty in transition and uptake illustrates the need for advancing work in what is considered 'DoD infrastructure'; the supportive technologies that enable the decision support and information awareness applications.

Working with users and having our work evaluated and assessed was invaluable. It helped us scope and refine our work and ensure that the research we were conducting would result in capability that actually satisfied operational need.

Despite the front-loaded effort to smooth transition, transition has been difficult. While the value has always been apparent to the research team, and various forms of demonstration, evaluation, and formal assessment have confirmed it, industry has been slow to adopt the technology. When questioned, industry has replied, "Where is the pain that this solution eases?" The answer to this question is explored in the next section.

Future Work

The missing component needed to stimulate industry adoption is 'consumer interest'. Industry is looking for 'user pain'; some indication that their investment in the technology will generate sales of their product and provide them with a return on the investment they make. Transferring intellectual property to the XSF and providing open source reference implementations lowers the cost of entry, but there will still be significant effort required to implement the technology in their respective products.

One method of generating this consumer interest is continued and wider exposure of the technology to DoD users. This can be done by participating in formal experimental venues, and the project has attempted to do so. The project was accepted to the Coalition Warrior Interoperability Demonstration (CWID) '08 but due to budgetary constraints the project was forced to withdraw. The project will continue to explore other avenues and other funding sources.

Once industry is convinced of the need, members of the project will work with industry to foster adoption of the technology and with the XSF to refine and further promote the standard. Project members will also work with members of, and system integrators from, the DoD and the IC to develop the interfaces necessary to support interaction between CDO-enabled chat and the Systems of Record they represent. For CDO-enabled chat to provide the benefits that we have been able to demonstrate in the lab, the Systems of Record must expose the data they contain and support automated machine-to-machine data transfers.

References

Defense Information Systems Agency. Net-Centric Enterprise Services (NCES) User Guide, Version 1.1/ECB 1.2. 20 April 2007. Available at http://www.disa.mil/nces/NCES_UG_Final_v1_1.pdf

Office of the Assistant Secretary of Defense, Networks and Information Integration, Information Management. DOD Discovery Metadata Specification (DDMS). Available at https://metadata.dod.mil/mdr/irs/DDMS/

Kelly, K., Wildes, F., Jacobsen, J., et al.: "Blue Force Tracking (BFT) Community of Interest (COI) Service: Reference Implementation Specification (v1.0)", MITRE Technical Report: MTR 04W0000109. Oct. 2004.

Organization for the Advancement of Structured Information Standards (OASIS). Common Alerting Protocol, v. 1.1. October 2005Available at http://www.oasis-open.org/committees/download.php/15135/emergency-CAPv1.1-Corrected_DOM.pdf

Winkowski, Daniel; Krutsch, Michael. 2007. Collaborative Data Objects. MITRE Technology Symposium. http://www.mitre.org/news/events/tech07/9.html#2686

Winkowski, Daniel; Krutsch, Michael, Bryson, Dave; Smith, Chad; Jasen, Jacobsen; Huss, Marshall. 2007. XEP-0204: Collaborative Data Objects. MITRE XMPP Standards Foundation XMPP Extension Protocol. http://www.xmpp.org/extensions/xep-0204.html

Krutsch, Michael; Winkowski, Daniel. 2007. Collaborative Data Objects. SPAWAR Joint Systems Chat Conference.

Krutsch, Michael; Huss, Marshall, 2006. FY06 and FY 07 open source reference system(s) SourceForge XMPPCDO project (**http://sourceforge.net/projects/xmppcdo/**).

Agarwal, S.; Starobinski, D.; Trachtenberg, A.; On the scalability of data synchronization protocols for PDAs and mobile devices. Network, IEEE Volume 16, Issue 4, July-Aug. 2002 Page(s):22 - 28

Benjamin C. Pierce, Alan Schmidt, and Michael B. Greenwald. Bringing Harmony to optimism: An experiment in synchronizing heterogeneous tree-structured data. Technical Report

MS-CIS-03-42, Department of Computer and Information Science, University of Pennsylvania, 2004.

SyncML specifications. http://www.syncml.org/downloads.html.

Collaborative Data Objects

Dan Winkowski
Michael C. Krutsch
757-825-8513 • winkowski@mitre.org
757-825-8510 • michael@mitre.org

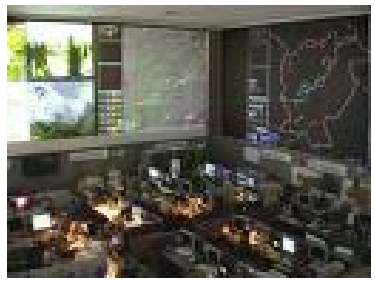
Mission Oriented Investigation Experimentation

MITRE Technology Program

This material was prepared under the FY07 Air Force MOIE Program. Approved for Public Release; Distribution Unlimited. 07-0974

Background






J. F. C. Fuller: "To establish a new invention . . . is like establishing a new religion—it usually demands the conversion or destruction of an entire priesthood."

Transformation to Net-centric Operations

General Problem

- **Collaborative environments (CE) are not cleanly integrated with applications or the enterprise**

Observations
Improving Time-Sensitive Team Decision Making: AF MOIE, Lindsley Boiney

- Both ADOCS and chat message indicating SAR imagery on a target now available. Operator is frustrated it doesn't specify the quality of that imagery: "Imagery of what? Is it *useful*?"

- TST Chief received information via chat regarding a Predator feed. He had to do a time-consuming back and forth on chat to find out which of 3 Predators it was referring to.

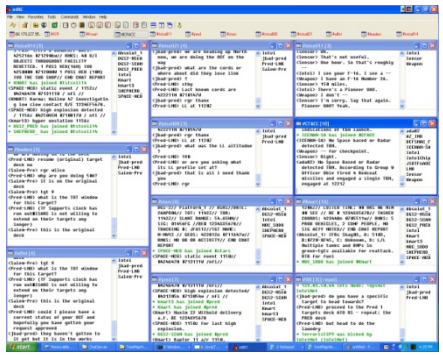
- "It is important to sort out what information really matters and to verify the source of the information before acting on it"

"Rubbish in, rubbish out – you've got to have a human in the loop when there's ambiguity."

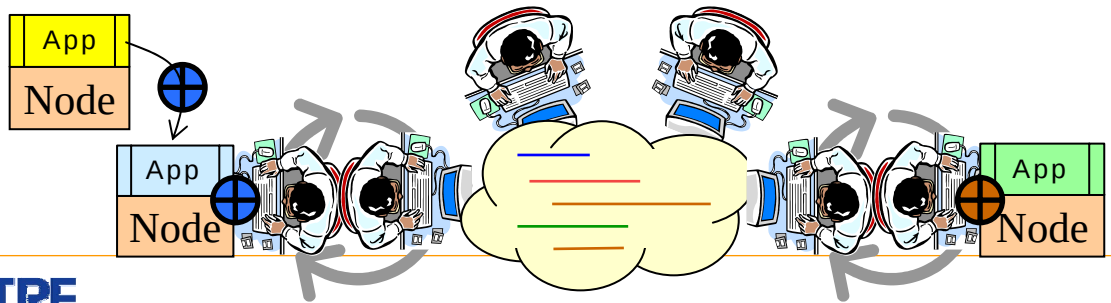
Humans have to sort it all out, put it in perspective, resolve inconsistencies, anticipate effects

Specific Test Cases

- **Chat and the Enterprise do not communicate**
  - **Enterprise has no visibility into chat spaces**
    - **Users spend a lot of time 'monitoring' chat to maintain situational awareness**



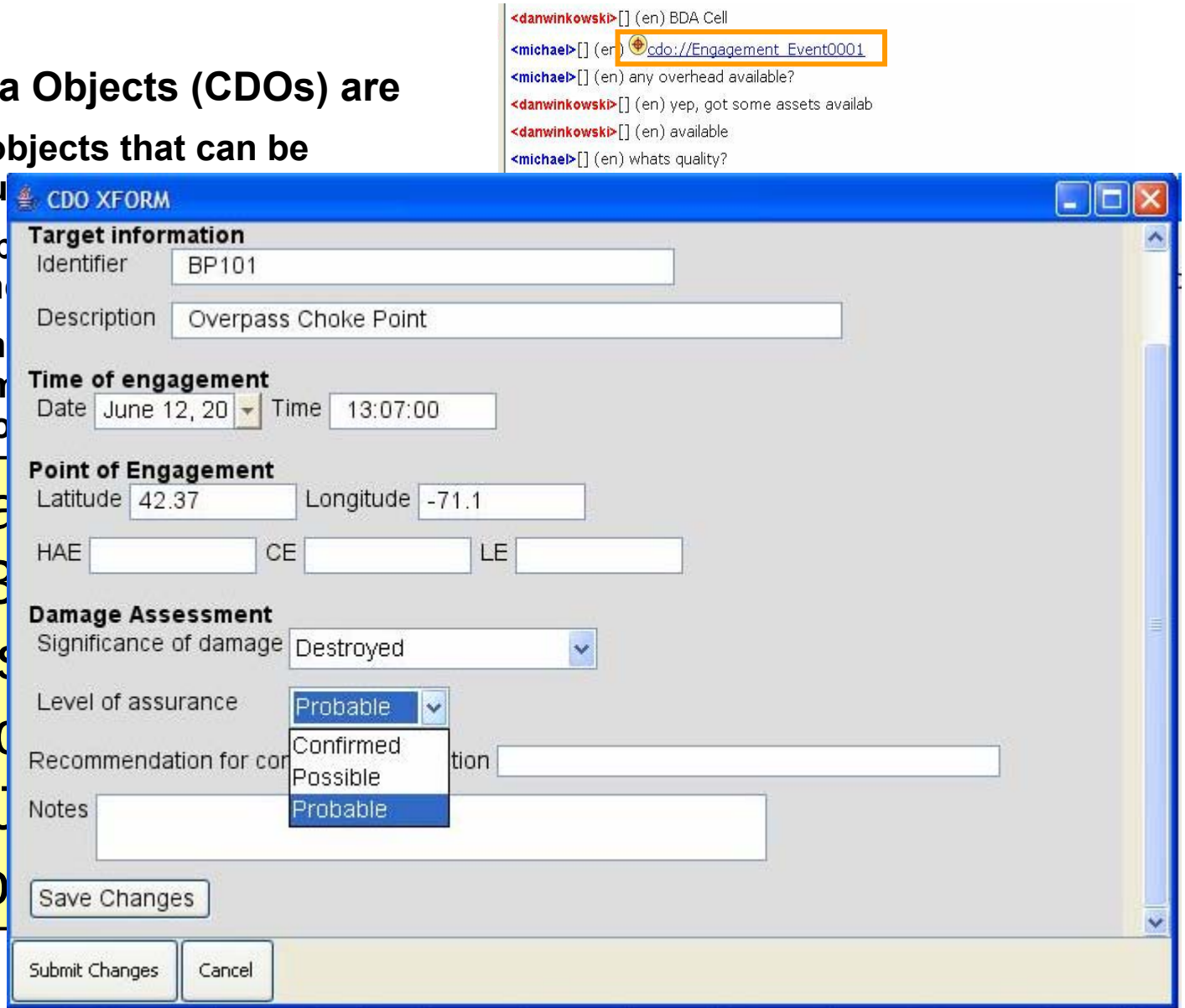
  - **Operators lose time and focus when they leave chat to interact with mission systems and enterprise capabilities in order to support collaborative work**
    - **Workflows and business processes are impeded by poor integration**



Approach

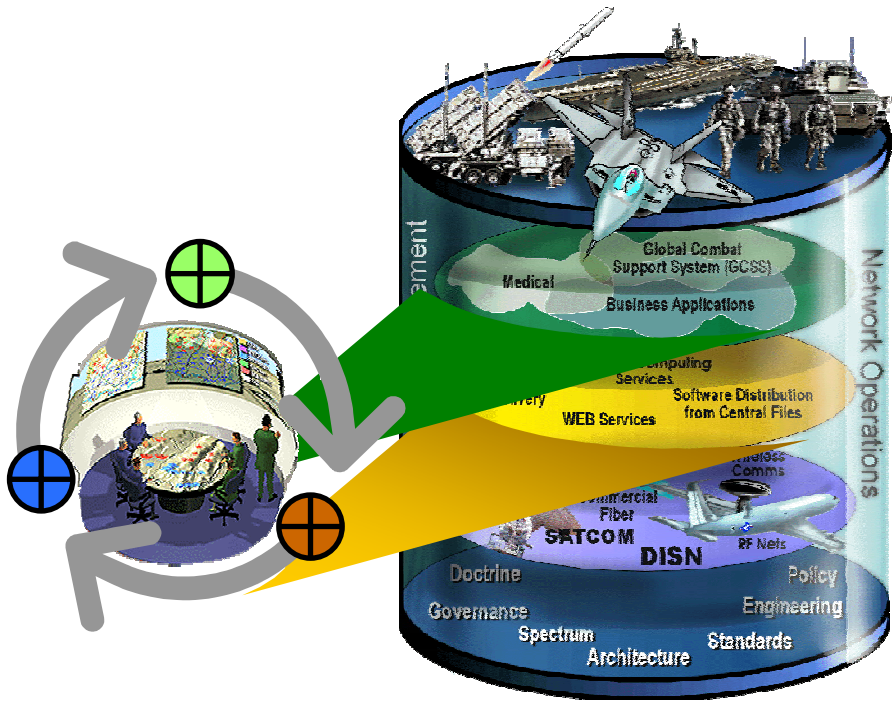
- **Collaborative Data Objects (CDOs) are**
  - **Smallish data objects that can be created/manipu**
  - **A natural data b applications an**
  - **A context for in web services, m support collabo**

"My criteria learn it in 3 not mimic s it is the wrc - Lt. Gen. J Multi-Natio



Capability Demonstration Overview

- Enhanced chat augmented with structured information encapsulated in collaborative data objects
- Net-Centric query of augmented chat spaces
- Chat/Enterprise integration via information services

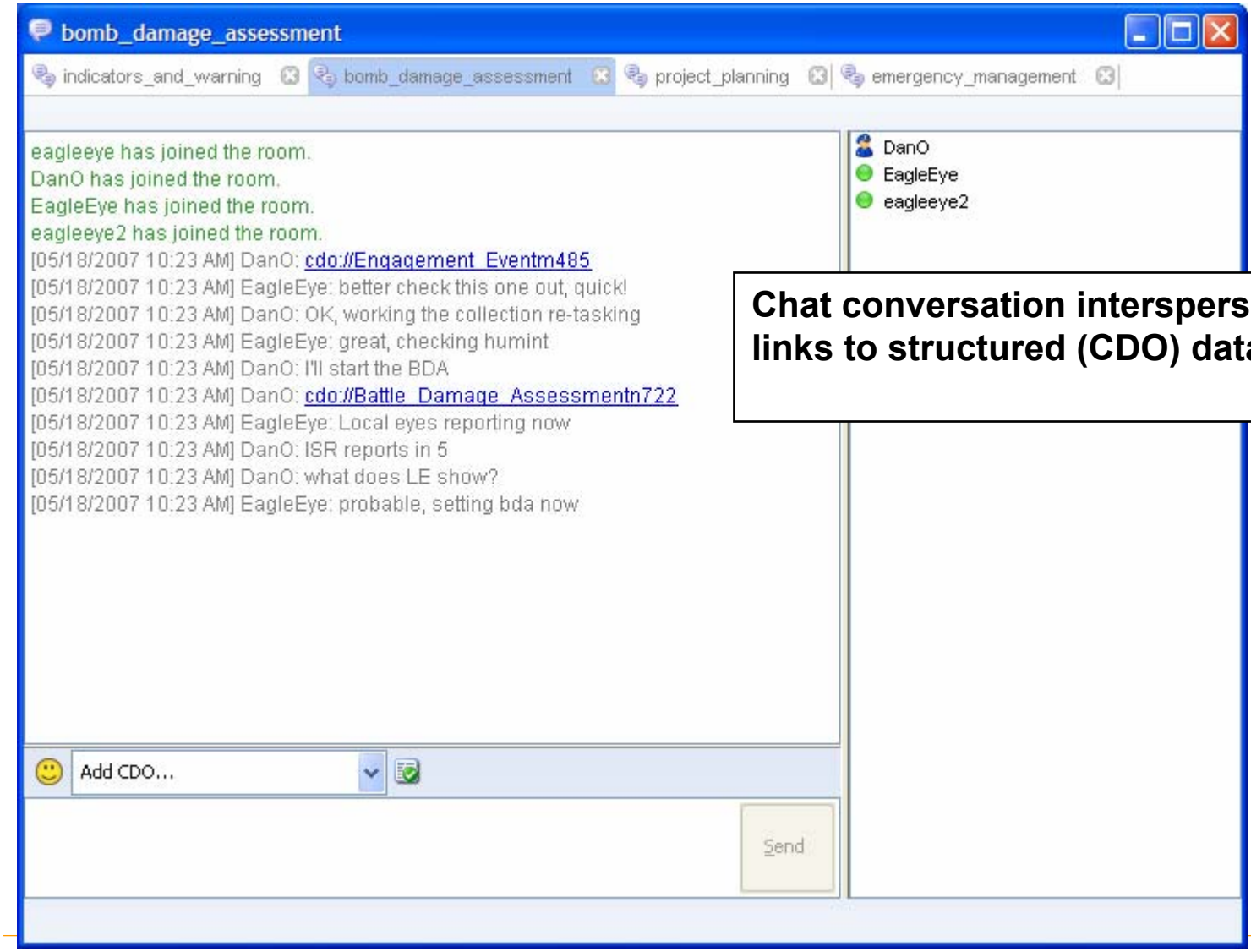


1) Chat Augmented With Structured Information

- Collaborative Data Objects (CDOs) Framework
  - Encapsulation of structured data linked to chat
  - Support for collaboration over CDOs
  - Synchronization protocol
  - Application/Chat interaction through CDOs
  - Description language for defining CDO types

- Operational impacts
  - Increased speed, agility, and quality of data focused collaborative decision making, SA, information production, and exploitation
  - Reduced ambiguity and operator overload

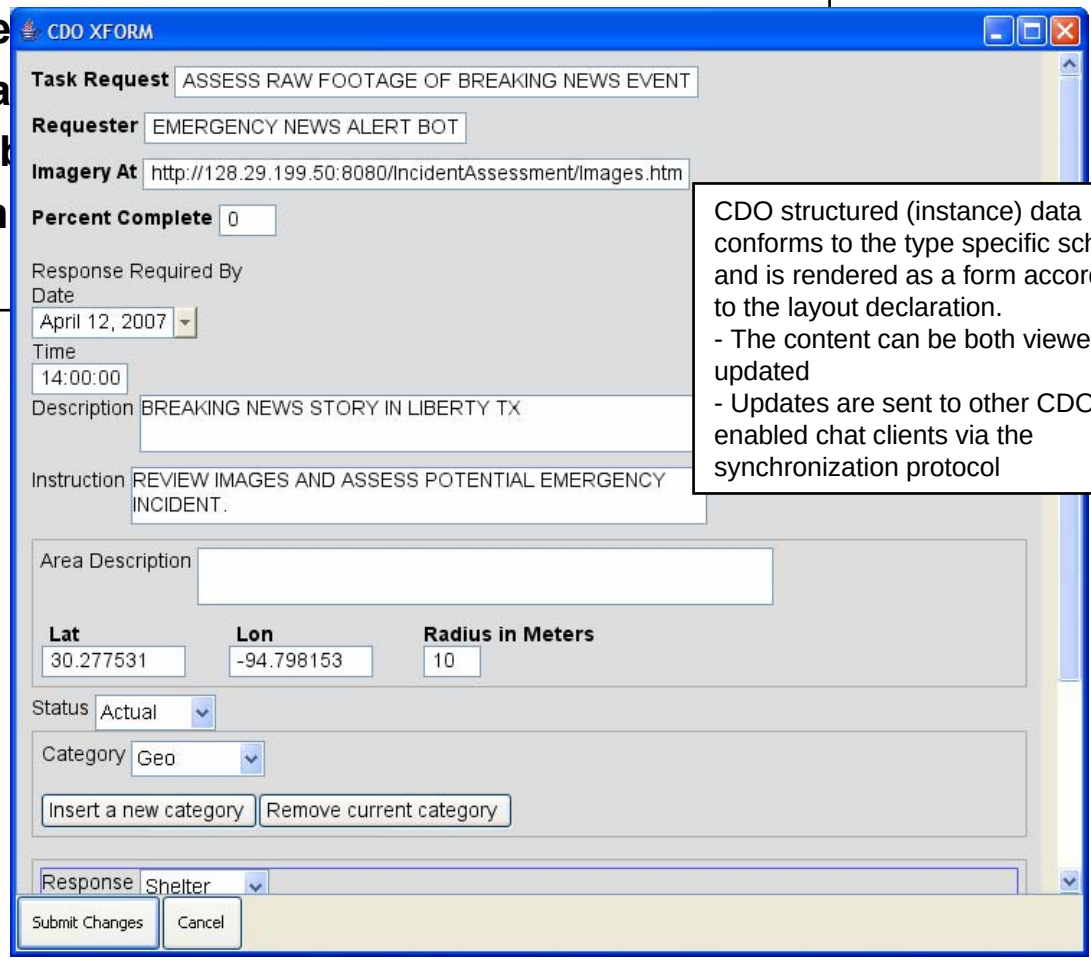
Chat Enhanced By Structured Data



Chat conversation interspersed with links to structured (CDO) data

Collaborative Data Object Type Definition Template

```
<cdo:Definition>
  <Metadata> label, version, de
  <Schema> W3C XML schema
  <Methods> Actions that can b
  <Layouts> W3C XForms com
</cdo:Definition>
```



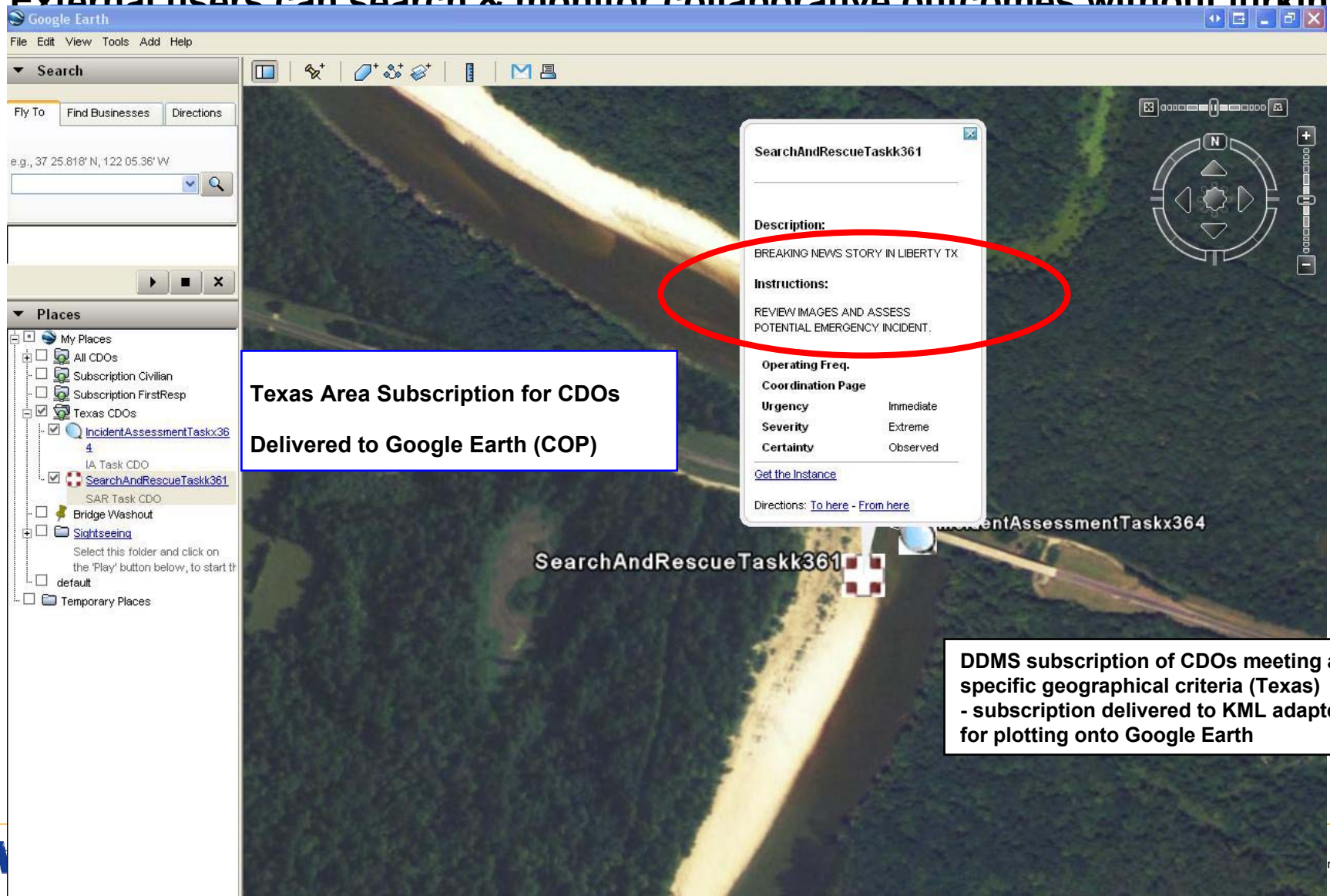
CDO structured (instance) data conforms to the type specific schema and is rendered as a form according to the layout declaration.
- The content can be both viewed and updated
- Updates are sent to other CDO enabled chat clients via the synchronization protocol

2) Net-Centric Content (Data) Discovery

- Enterprise visibility into CDO augmented chat spaces
  - CDO Advertisements (per DDMS)
  - Query over collaborative work products (NC Content Discovery proxy)
    - Link to Publication/Subscription mechanism (DDS)
  - Syndication of CDOs within chat rooms (RSS)
- Chat user visibility of enterprise content
  - Special “Query” CDO type
  - Supports enterprise content discovery from within chat
- Operational impacts
  - External users can search and monitor collaborative outcomes without lurking in chat rooms
  - Chat users can directly query enterprise information assets

Advertise & Subscribe: Provide the Enterprise With Access to Chat Products

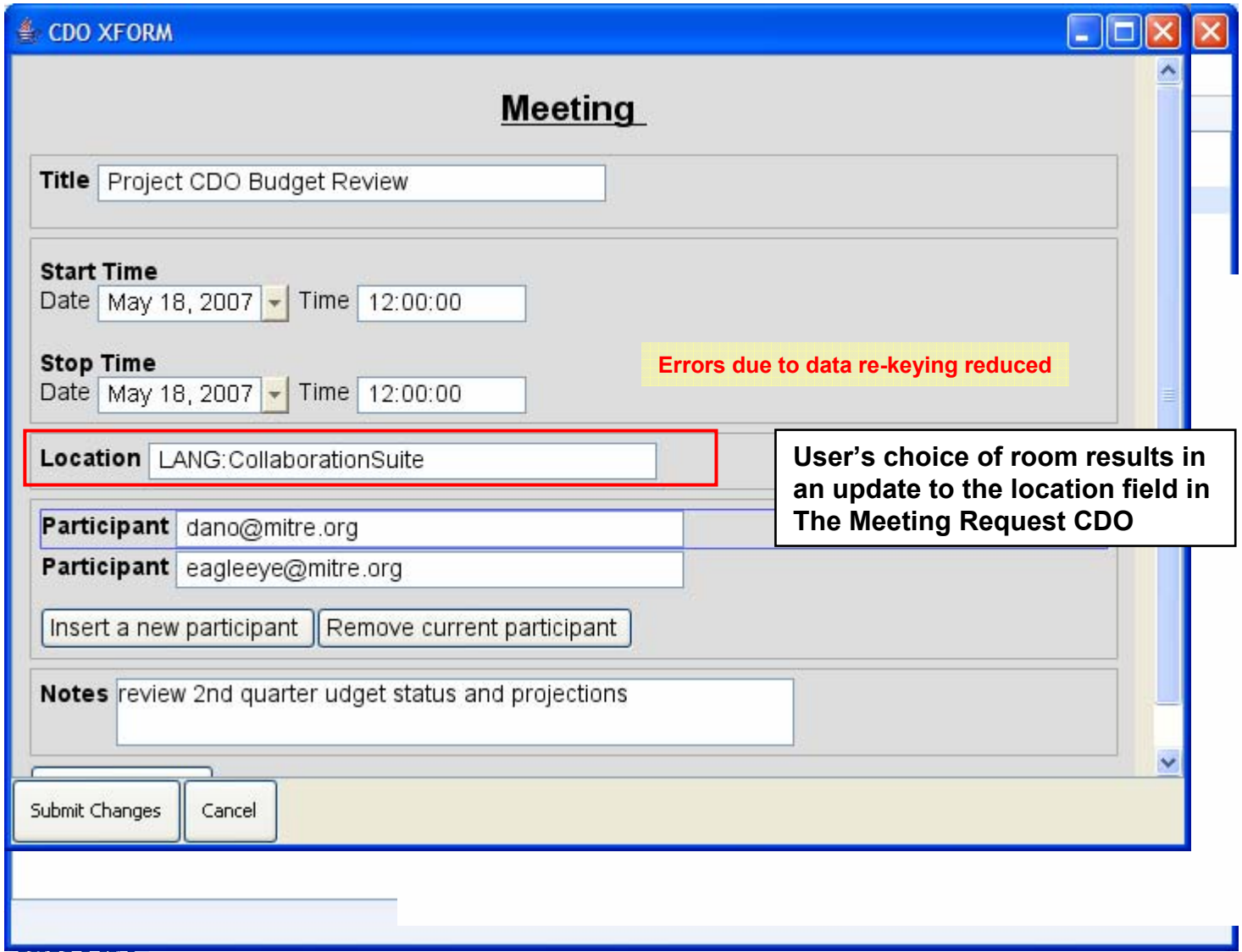
- External users can search & monitor collaborative outcomes without lurking



3) Chat/Enterprise Integration via Information Services

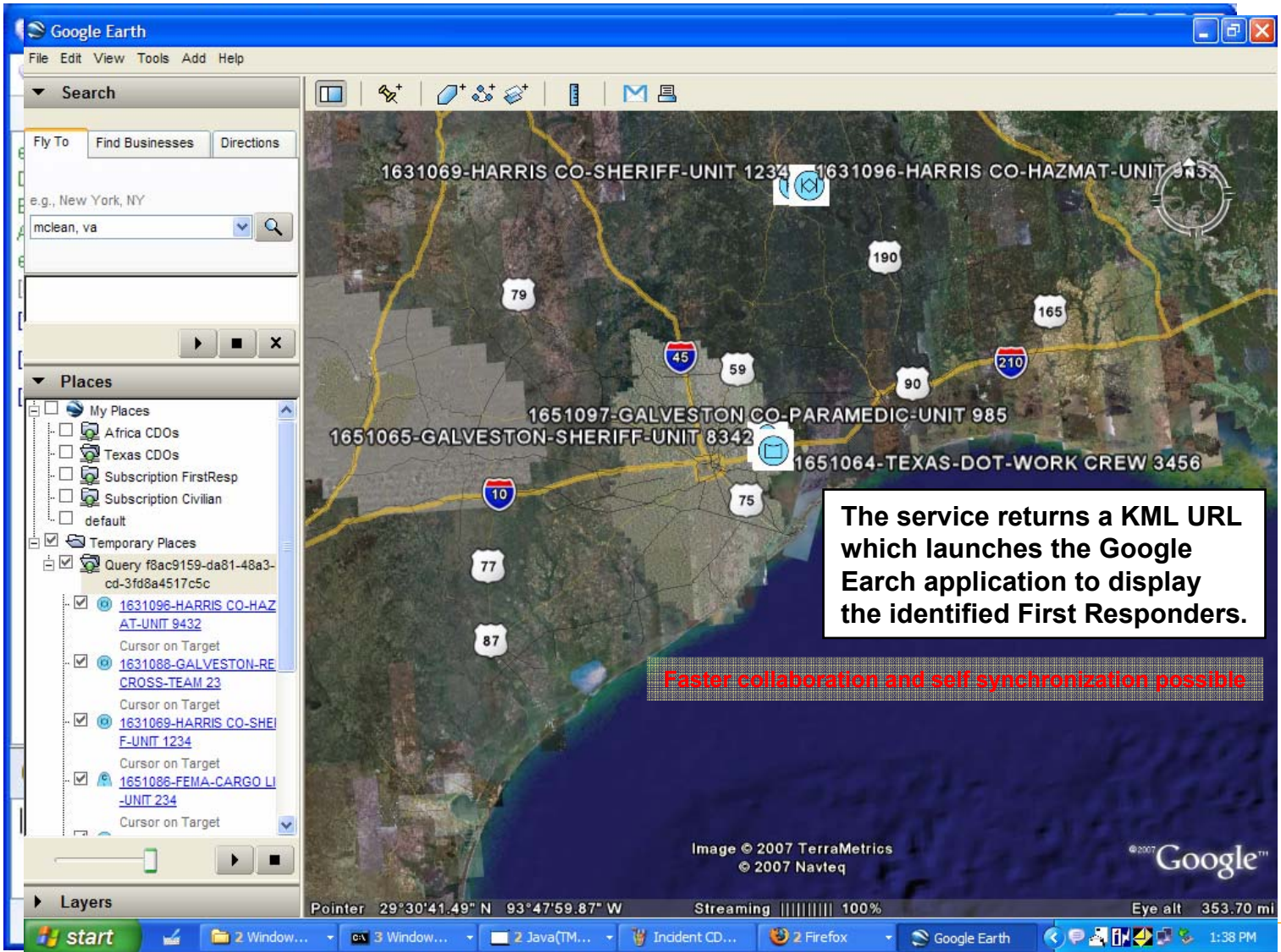
- Operators can directly access enterprise information services according to the mission context (via CDO typing)
  - Cognitive disruptions minimized
  - Errors due to data re-keying reduced
  - Faster collaboration and self synchronization possible

Chat Operators Can Directly Access Relevant Enterprise Information



Example 1: Identify resource availability per criteria in CDO fields and bind result to another CDO field – **Find meeting rooms at date/time to accommodate N participants**

Chat Operators Can Directly Access Relevant Enterprise Information

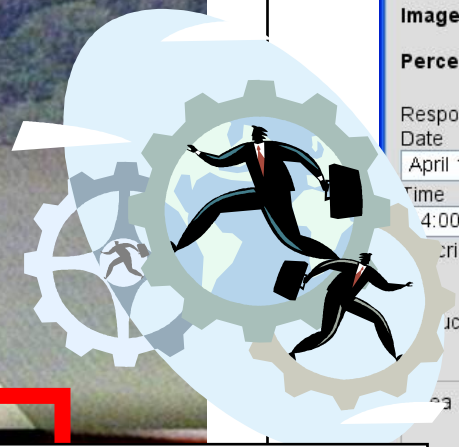


Example 2:
Pass parameters to an application and request an external action to be performed – **Plot first responder units in the vicinity of a SAR location**

HIT (Human Intelligence Task): Software Calls People as a Service

- Mission workflows are speeded through novel application of operator expertise and familiar tools
  - Mission services can directly request expertise resident in chat rooms
  - Chat rooms become *in effect* an enterprise information service

News Monitoring Application



CDO XFORM

Task Request: ASSESS RAW FOOTAGE OF BREAKING NEWS EVENT

Requester: EMERGENCY NEWS ALERT BOT

Imagery At: http://128.29.199.50:8080/IncidentAssessment/Images.htm

Percent Complete: 0

Response Required By Date: April 12, 2007

Time: 4:00:00

cription: BREAKING NEWS STORY IN LIBERTY TX

uction: REVIEW IMAGES AND ASSESS POTENTIAL EMERGENCY INCIDENT.

a Description:

at: 0.277531 | Lon: -94.798153 | Radius in Meters: 10

tus: Actual

ategory: Geo

sert a new category | Remove current category

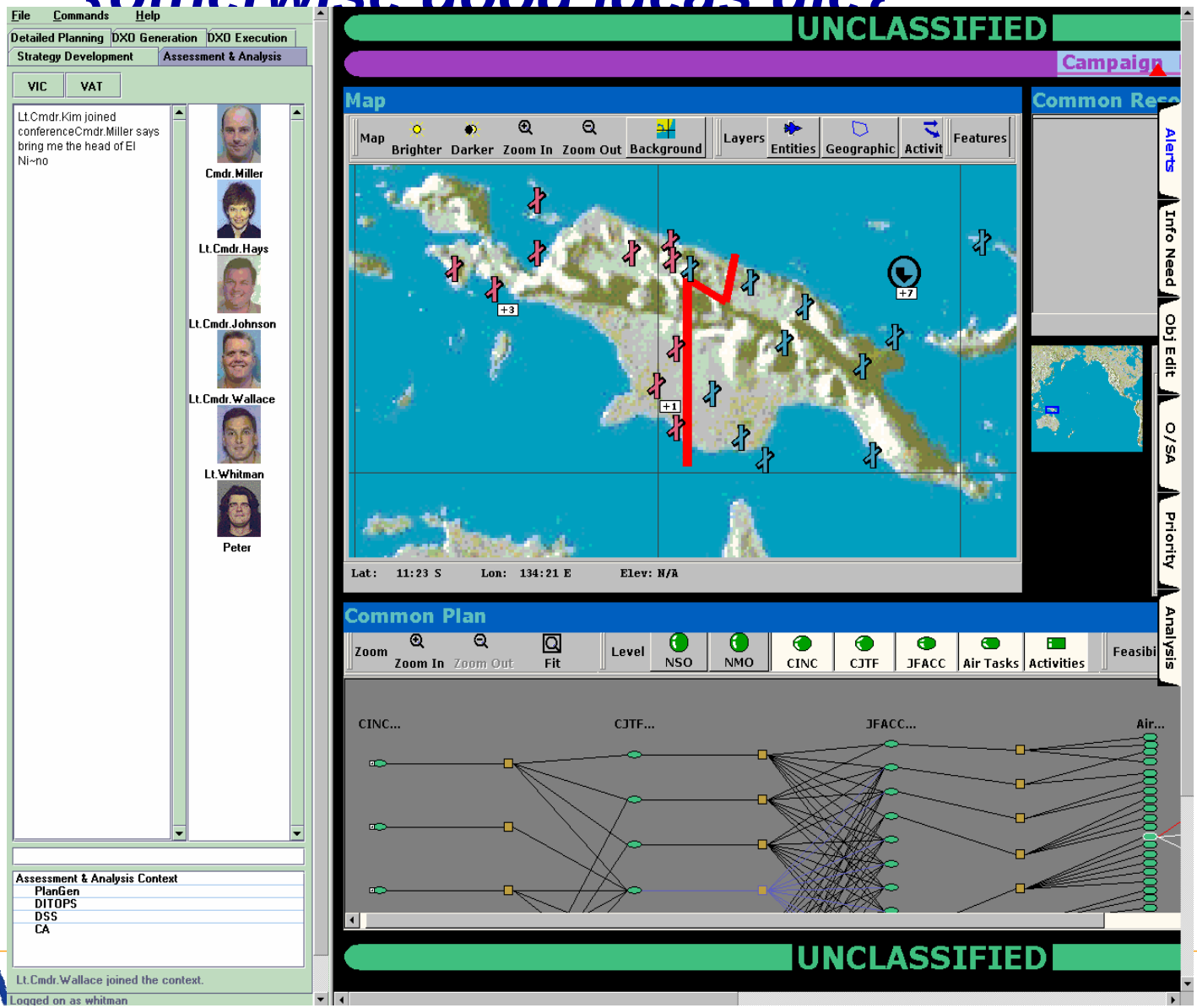
Cognitive disruptions minimized

►News Monitoring application injects an Incident Assessment task into the chat room.
◄ After viewing the video the form is completed and the structured data returned to the calling application.

Chat Operators Can Directly Access Relevant Enterprise Information

- Operators can directly access enterprise information services according to the business context (via CDO typing)
  - Cognitive disruptions minimized
  - Errors due to data re-keying reduced
  - Faster collaboration and self synchronization possible
- New technology developed to support chat/enterprise integration
  - CDO method description language supports a declarative, pattern based approach for describing CDO interaction with an information service
    - Addresses user input, method call type, service result types, data transformation, and output handling within chat
    - Plug and Play - no client modifications required to add methods
  - Developed a CDO Method Invocation and Binding Framework enabling enterprise service invocation and response handling per the method description language
    - Generosity promotes loose coupling, service endpoints may vary
  - Chat is positioned to participate in a SOA Enterprise

Transition Is Important
{*otherwise good ideas die*}

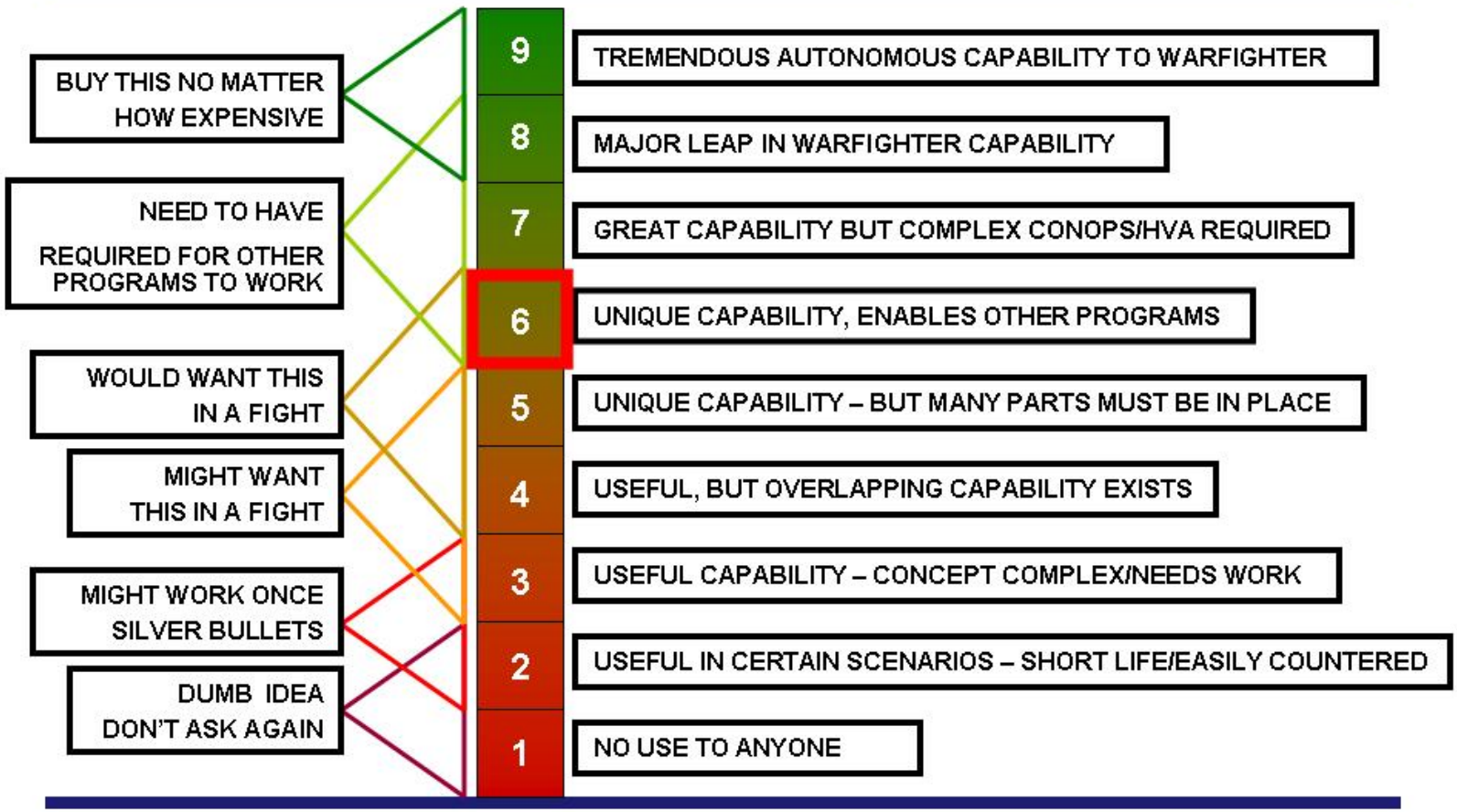


DARPA

Transition

CDO DA - Operational Value (A3)

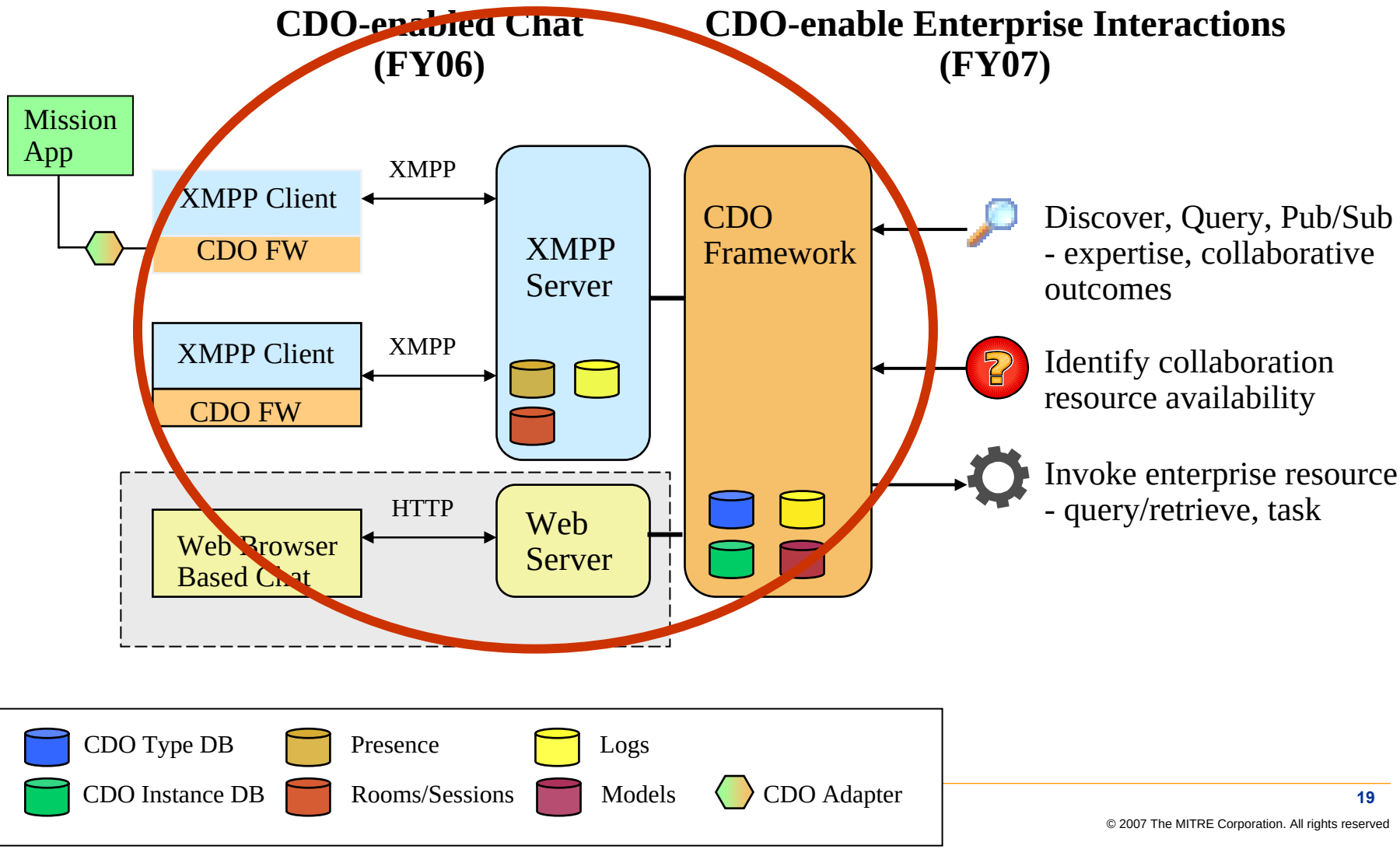
U.S. AIR FORCE



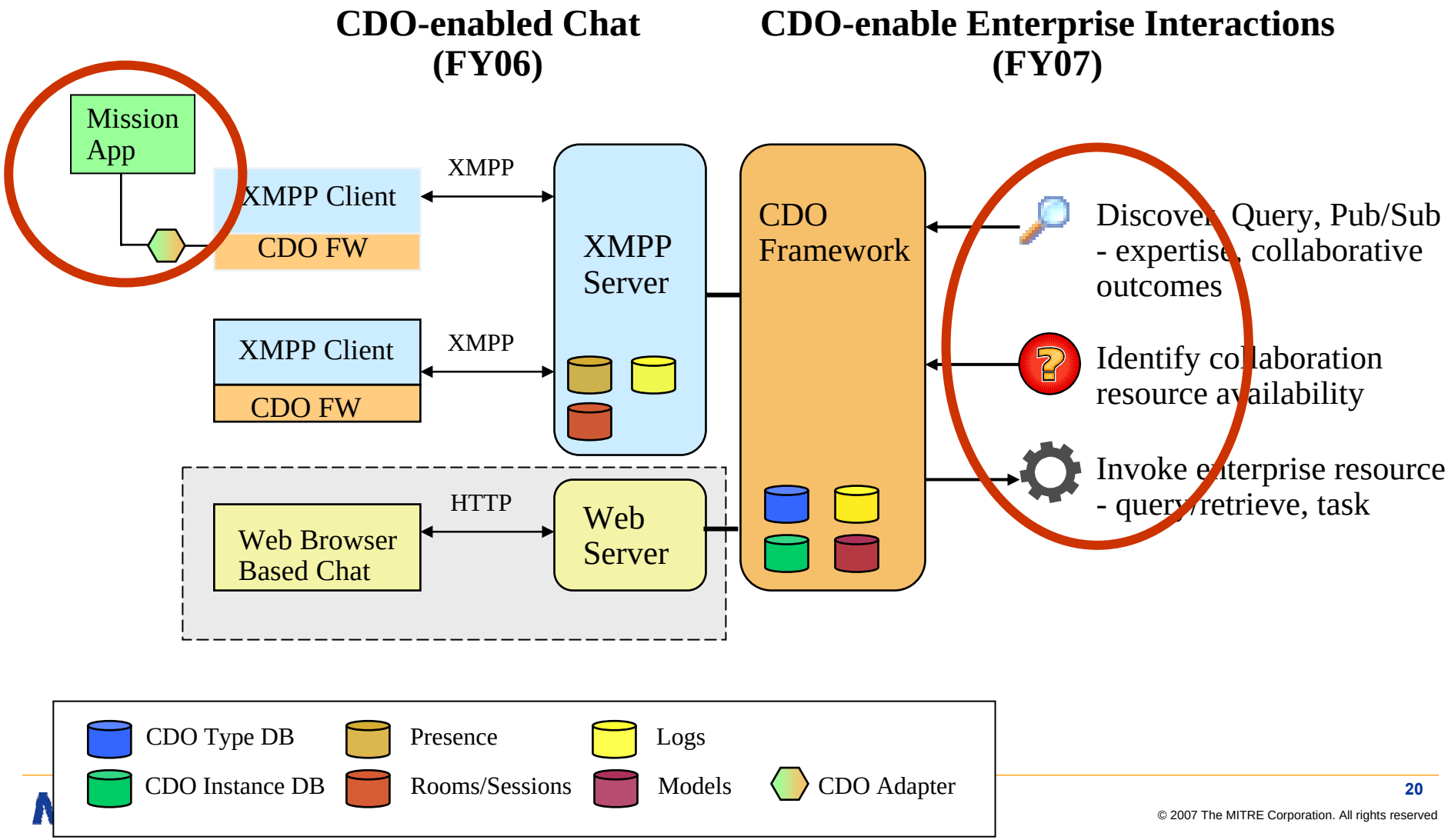
Tasks

software has
s ©2007 The

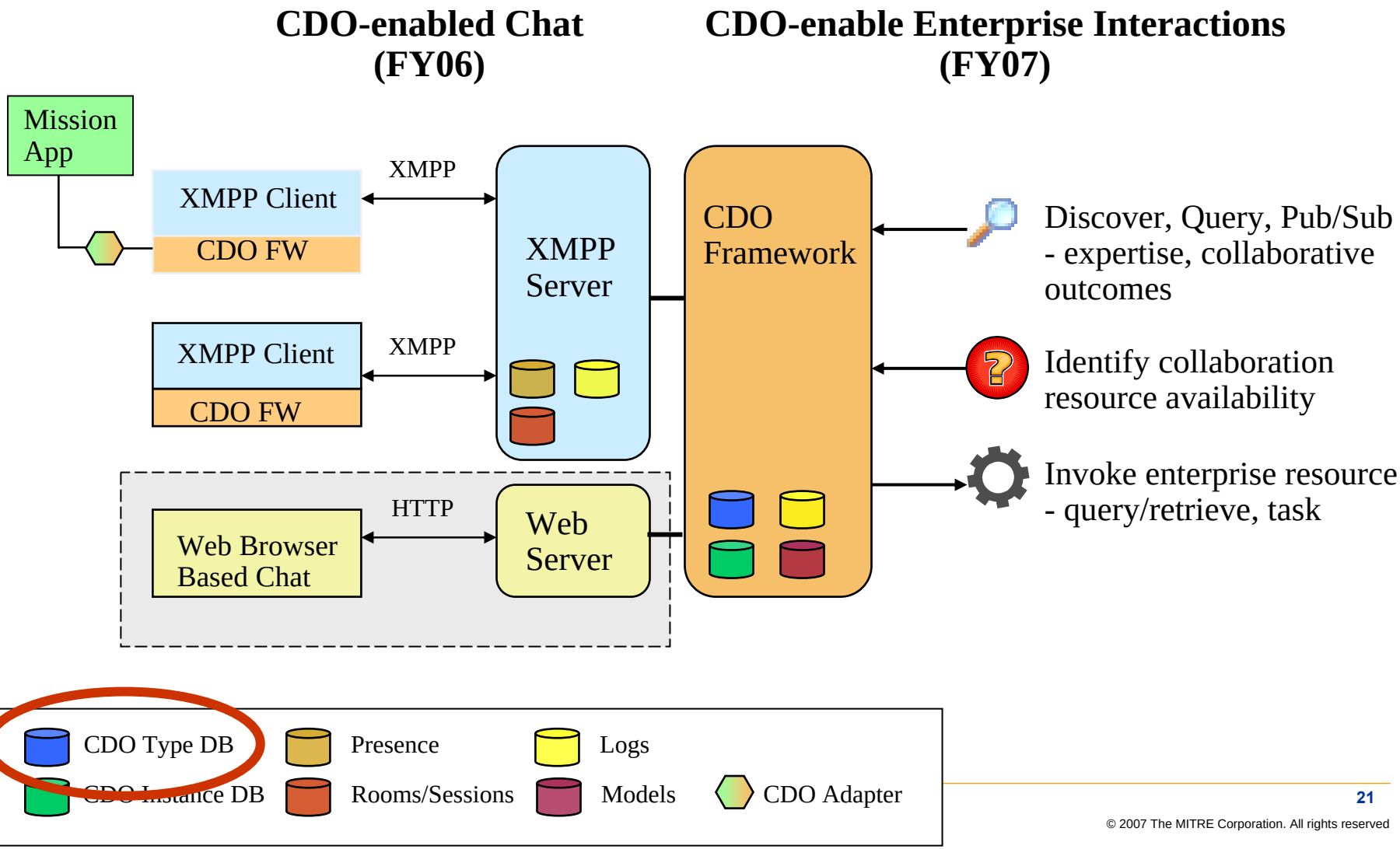
Collaborative Data Objects
XMPP Vendor Transition Opportunities



Collaborative Data Objects System Integrator Opportunities



Collaborative Data Objects
End User Responsibilities

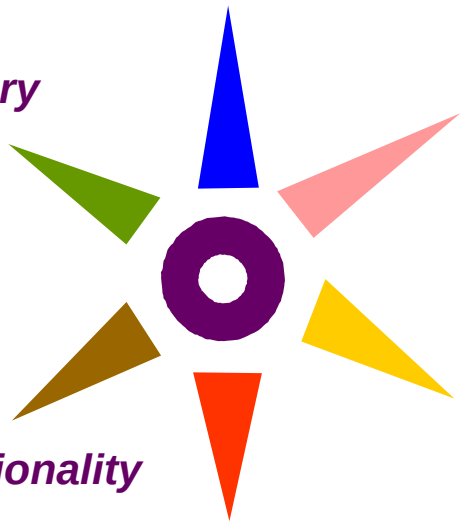


Collaborative Data Objects: Summary

In a nutshell, a Collaborative Data Object (CDO) is a(n)…

way to reduce the ambiguity of chat through an increase of structure, data quality and fidelity - follows the OHIO principle: Only Handle Information Once

basis for enterprise discovery of collaboration state, outcomes, expertise, availability, …



way to record decision making/coordination outcomes during collaboration

invocation point to access enterprise/application functionality

context for agent participation during collaboration

means for applications and the enterprise to inject structured data into the collaborative process and receive structured data in return

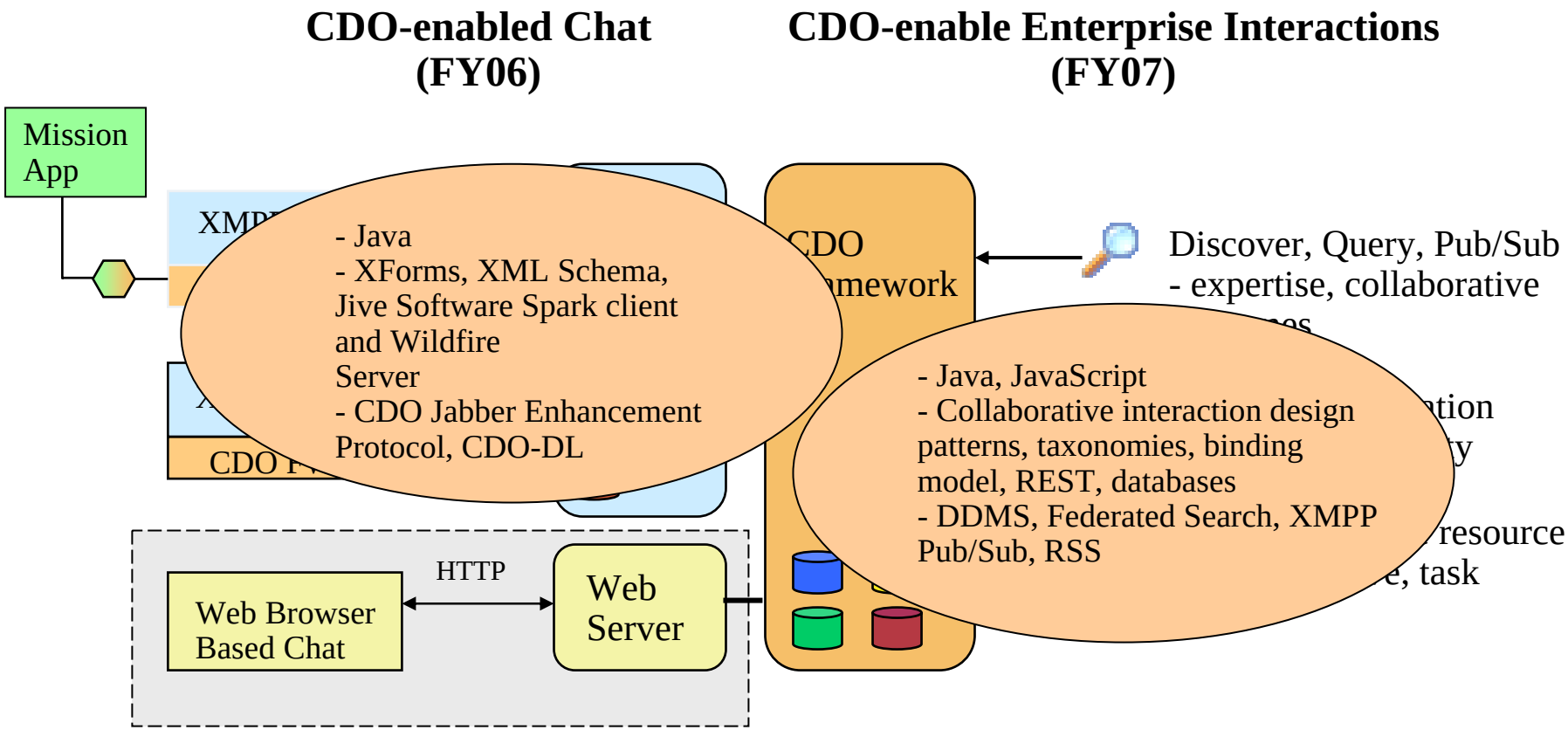
FY06
FY07

The End
(of the presentation)

Backups

- Technology Overview
- Interaction Patterns
- Accomplishments Summary

Collaborative Data Objects (Technology)



CDO Method Interaction Patterns

- UI Input Patterns
  - Menu visibility when parameter preconditions met
  - Parameter prompt
  - Method call or cache access
- Result Types
  - Content Reference : URL, CDO
  - Discrete Primitive XSD: Simple Type
  - List
  - Tree
  - Complex type
    - Binary (mime-type handling)
    - Structured data
- UI Output Patterns
  - Browser Display of URL
  - Screen Echo
  - Confirm/view dialog
  - List selection (single, multiple)
  - Tree navigate, select leaf
  - Tree navigate, select branch
  - CDO reference
- Routing Patterns
  - Transform
  - Transmit - send to chat as text
  - Negotiated method refinement
  - CDO item update/create/delete
  - Store locally to named cache
    - ID
    - methodID
    - CDO ID
    - Timestamp
- Method Invocation Patterns
  - Discover properties of identifiable resource
    - E.g. conference rooms in a facility
  - Range restriction
    - Where, When (P-Cot example)
  - Property value retrieval
  - Property value set

Accomplishments

- Designed the CDO IM architecture and framework
- Description language to define CDO Types

```
<cdo:Definition>
  <Metadata> label, version, description </Metadata>
  <Schema> W3C XML schema for the CDO </Schema>
  <Methods> Actions that can be invoke on a CDO </Methods>
  <Layouts> W3C XForms component description </Layouts>
</cdo:Definition>
```

- Published CDO XMPP Extension Protocol (XEP-0204)
- Enabled Net-Centric query of CDO augmented chat spaces
- Developed Chat/Enterprise interaction models
- Posted and open sourced a reference implementation
- Evaluated effectiveness through operator forums